Revitalizing Indigenous Languages

The Native Languages Revitalization Resource Directory

An Interactive Qualifying Project Report

submitted to the Faculty of Worcester Polytechnic Institute in partial fulfillment

of the requirements for the Degree of Bachelor of Science

Submitted By

William Grebe

Kevin Janesch

Susan Stukas

Submitted To

Fabio Carrera, Ph.D

Joseph Farbrook, Ph.D

Abstract

The goal of this project was to work with the Indigenous Language Institute in Santa Fe, NM and provide accessible, up-to-date resources that will help revitalize endangered indigenous languages. This was done by creating a digital resource directory website in MediaWiki format to host informational pages for language programs, funding sources, and coordinating organizations. In order to create relevant up-to date pages, the team contacted organizations, sent out surveys, and confirmed existing information. The team wrote programs to automatically convert the updated information to wiki-pages, and created a mechanism for organizations to keep their information updated and available.

Executive Summary

For this project, we worked with the Indigenous Language Institute in Santa Fe, New Mexico to help revitalize endangered indigenous languages in North America. It is estimated that upwards of 90 percent of the 6,900 languages spoken in the world today will be extinct at the end of this century. The widely popular languages, such as English and Spanish, are pushing the indigenous languages out of homes, making it harder for the younger generations to learn the languages from their elders.

Substantial efforts have been made to help the younger generations learn their indigenous languages. The United Nations Organization for Education, Science and Culture (UNESCO) supports policies, collects data, and assists programs to strengthen their revitalization techniques. Another widely popular organization is the Endangered Languages Project, an online resource to record, access, and share samples of endangered languages. Local efforts include the Indigenous Language Institute (ILI); the ILI researches, teaches and shares information and tools to empower Native communities to make their languages seen and heard everywhere and by everyone. Their most prominent contribution is the Native Languages Revitalization Resource Directory, published in 1998.

While the Resource Directory of the ILI was an extraordinary endeavor, it has been approximately fifteen years since the directory has been updated; it also has limited reach as it is in a paper-only format. Additionally, many of the resources listed have changed names, contact information, and programs offered.

By completing this project we hope to give the ILI a new and improved digital resource directory that will be of great help in revitalizing indigenous languages across North America. To accomplish this, we have updated the current Native Languages Revitalization Resource Directory. From there, we transferred the updated entries into our final product: a Wikipedia-like directory accessible from the ILI's website that has the same intent as the paper version, with each resource having a dedicated Wiki page. We have also created a variety of ways to ensure the maintenance of these Wiki Pages.

To update the current online resource directory, we scoured the Internet and found as much as possible; mainly contact information. Subsequently, we used that updated contact information to reach out to the various resources via email and attach an online questionnaire complete with fields to write down program information, organization information, and anything additional that they
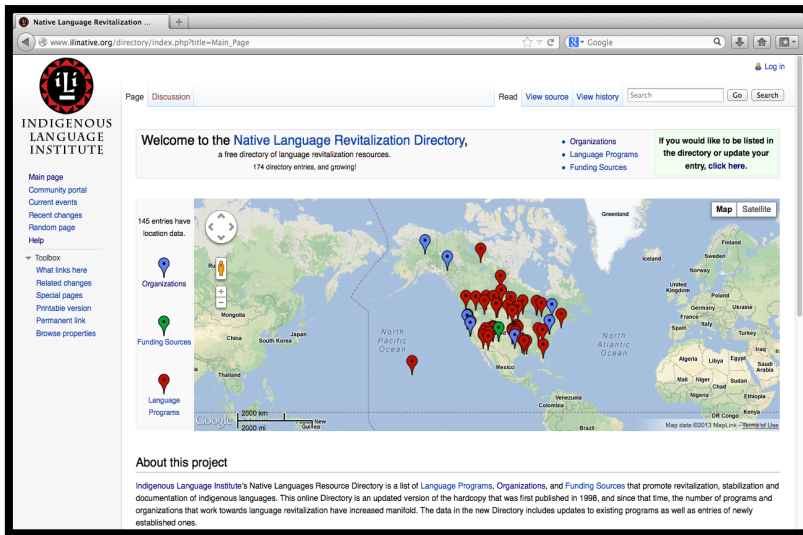
would like to be included in their directory entry. As they responded to the questionnaire, we stored the new information alongside the already-existing information in our database. We then created a

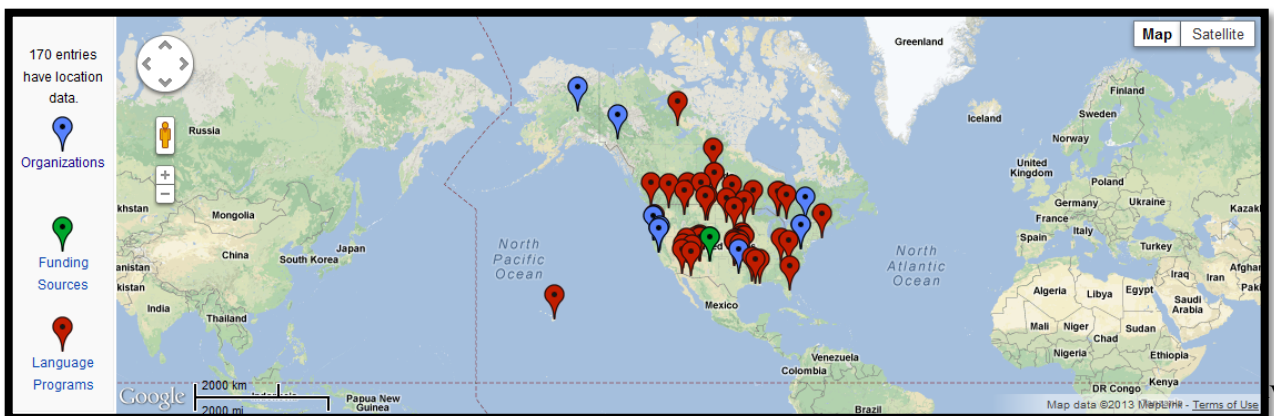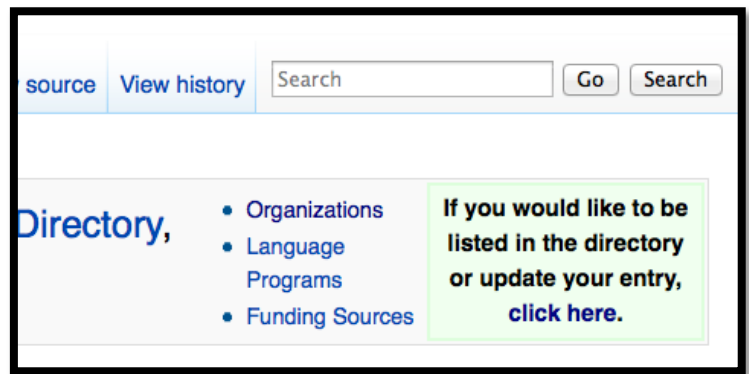Homepage of the Native Language Revitalization Directory

The Native Language Revitalization Directory that we have created is an improvement of the one currently in use because of its accessibility, searchability, and perpetuity. Our project can reach a wider audience because placing the information online has lowered the access barrier. The home page of the website as shown to the left has a few different features that assist the searchability of the directory. The first way to search is the standard search bar in the top right corner of the page. You can use the search bar to easily find a particular query of your choosing, such as a specific language. Another simplified way to search is to use the three categories listed just under the main search bar. These categories, when clicked on, will bring up a list of alphabetized associations based on the category that was selected. Further, we have included an interactive map that has pins color-coded by category of Organizations, Funding Sources, and Language Programs.

Because the directory is now online and is accessed more easily by people, the need for accurate information is greatly increased. To combat this need, we provided helpful materials that relate to updating and maintaining the Resource Directory. These materials include text tutorials for copy and pasting specific Wiki syntax, video tutorials to show basic navigation of the editing process, and an article wizard to filter out information that does not need to be on the Resource Directory.

We made these enhancements as significant as possible to ensure that the new and improved version fit in with the modern concept of Internet use, as well as having a profound impact on communication between the language revitalization communities. A noteworthy amount of consideration and evaluation went into the design of this directory, and both its attributes and structure reflect the quality of the directory. This Online Directory will be of substantial benefit for both language programs and people to connect to each other and share critical information for language revitalization.

Table of Contents

1. Introduction

Of the 6,900 languages left in the world, studies predict that anywhere from 60 to 90 percent of them will no longer be spoken at the end of the century (Hinton, 2008). This decline is due to a number of factors, the most significant being the fact that these languages are not being spoken as a first language. Another key factor of language endangerment is that indigenous languages are not being passed down to the younger generations. A language becomes endangered based on its number of speakers, the age demographic of those speakers, and the domain in which the language is spoken (Woodbury, 2012).

North America is home to 629 indigenous languages. Canada holds 86 of these, all of which still have native speakers. In Mexico, there are 298 indigenous languages listed; 291 of those are living languages and 7 are no longer spoken. Out of the 245 documented indigenous languages in the United States, there are 176 living indigenous languages; 65 languages have succumbed to extinction, and 4 languages have been relegated to being second languages (Ethnologue, 2013). Most of these living indigenous languages have been in existence for centuries, and future predictions show that within the next century these

Figure 1: Indigenous Languages in North America

languages will be extinct. Specific to the Santa Fe area, there are five indigenous languages. These indigenous languages are Zuni with 9,686 fluent speakers, Keres with 8,580 fluent speakers, Towa with 3,000 fluent speakers, Tewa with 1,500 fluent speakers, and Tiwa with about 2,625 fluent speakers (Endangered Language Project).

Indigenous languages are endemic to small areas and have no national status anywhere, nor is there a designated location to learn their ancestral tongue (McCarty, 2003). This loss of language is significant because language is used to form ideas, theologies, environmental knowledge systems (i.e. botany, zoology, etc.) medical knowledge, cultural practices and artistic skills, in addition to many others (Hale, 1999). Arguments can be made to translate everything into one language; however, many connotations can get lost in translation. Each language represents another lens to look at the world through. A multitude of strides have been made to preserve these languages, most notably by the Indigenous Language Institute.

The Indigenous Language Institute was founded in 1992 as the Institute for the Preservation of the Original Languages of Americas, more commonly referred to as IPOLA. To expand their services to all of the indigenous communities, in 1997 they changed their name to the Indigenous Language Institute (Indigenous Language Institute, 2009). Over the past decade, the ILI has developed a directory of organizations and programs for those that want to keep a language alive. To some degree, this directory has been maintained over time and has recently been moved to the ILI website. However, there is no efficient way of searching through this website when looking for a specific language or resource, such as private funding, organizations, individual scholars, and so forth. In addition to the resource directory, the ILI has created a technological learning space, as well as sessions to teach people the endangered languages around Santa Fe. They also provide references and materials to create multimedia sources to perpetuate this learning. The ILI has demonstrated strong efforts in building partnerships with other organizations to help further their progress. In order to continue to solidify the strong foundation that is currently established, the Indigenous Language Institute is interested in updating their resource directory.

With this project, our group will fill this technological disparity by updating the resource directory, as well as providing a user interface that will enable people to use the new and improved resource directory in an efficient manner. By updating the resource directory, providing a searchable website to host those resources, and creating training materials for community members to learn the process of placing their knowledge and information to the website, we intend this to be of great assistance to the Indigenous Language Institute in their efforts of revitalizing indigenous languages. These contributions will enable the native speakers of their respective tribes to find resources to pass on their language to future generations while simultaneously preserving their culture and traditions.

2. Background

All languages, be they spoken or written, embody the culture from which they originated in the world. Across the globe, there are a vast number of languages; each of which has their own identity linked to them. In this section, discussed will be the languages of the world and the plight of the small language in modern-day society, the endangered languages both globally and specifically within Santa Fe, New Mexico. Also discussed will be the various classes of endangerment, with a clear outline of the levels, as defined by the United Nations Educational Scientific and Cultural Organization (UNESCO). Through partnerships with indigenous peoples, UNESCO seeks to "support them in addressing the multiple challenges they face, while acknowledging their significant role in sustaining the diversity of the world's cultural and biological landscape" (UNESCO).

2.1 Endangered Languages in the World

The world is filled with many interesting cultures, and behind each of these cultures is a unique language; these languages form the backbone of their traditions. Everyone uses some form of language to communicate with others. Language, adjacent with culture, dates back thousands of years. In more recent times, there has been a developing need for cross-communication through cultures and countries. International businesses and collaborations are becoming commonplace, creating a global environment and thus generating a need to adopt a common language. The desire to communicate in the world has caused many smaller languages to become less used in favor of languages spoken by a plethora of people, while expanding the base of speakers in already popular languages.
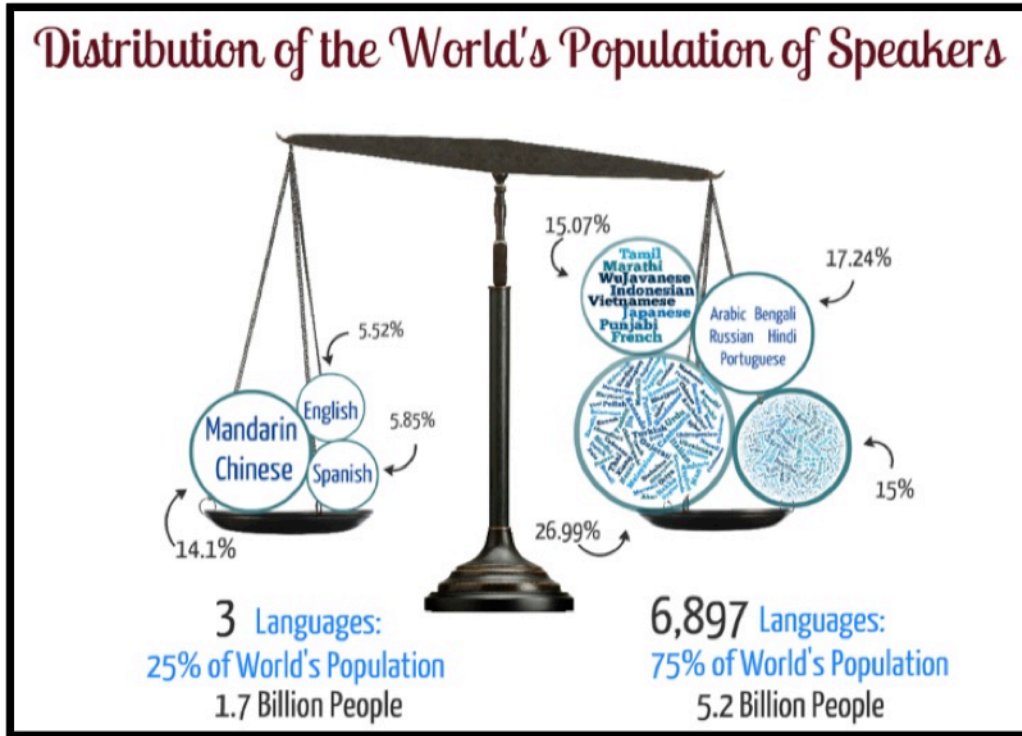
Figure 2: Distribution of World Languages by Population

As shown in the graphic above, the top 100 languages are spoken by approximately 85% of the world's population; while a mere 15% of the world's population speaks 6,800 of the world's languages. The three most popular languages, Mandarin Chinese, English, and Spanish, are spoken by a quarter of the world's population. A zoomed in view of the right side of the balance is shown below, which allows the viewer to note that seemingly popular languages (such as Polish or Italian) are not as widely spoken as imagined. The circle with a total of 17.24% listing Arabic, Bengali, Portuguese, Russian, and Hindi each account for roughly 2-4% of the world's population of speakers. The 15.07% circle denotes the languages that are each spoken by between 1 and 2% of the world's population. The largest circle, accumulating a total of 26.99%, is made up of 84 languages that all have less than 1% of the world speaking each of their languages.

There is currently pressure on smaller, indigenous languages from more popular languages. When natives first populated areas, communication between groups was poor, causing the languages spoken to differentiate from one another to the point that they became dialects or unique languages (Abrams, 2003). As long-distance communication became easier and communication in one's immediate area became necessary, one language would become dominant over the others. It would

then place pressure on the smaller, non-dominant languages to either continue about their business, largely isolated from their surrounding area, or learn the popular language (Reyhner, 1999). Many chose the latter, and more are choosing to do so now as things, such as using the Internet, become ever more prevalent and necessary skills; however, technology still lacks adequate translations or accommodations for these native languages. In addition, many of the indigenous traditions do not allow writings of their languages (Jones, 2013). For example, the Towa language, spoken by the Jemez Pueblo, has no written form because it would violate the tribal rulings. "Jemez is the only culture that speaks this language, and our traditional law forbids our language from being translated into writing in order to prevent exploitation by outside cultures." (Pueblo of Jemez). Because of this, it creates a higher level of difficulty to teach and learn these languages; it is also increasingly more difficult to keep the attention of younger generations that are more easily immersed in American culture. "Depending on attitudes toward the ancestral language, those children or their children may never learn the smaller language, or they may forget it as it falls out of use" (National Geographic, 2012).

2.1.1 Stages of Endangerment

This dearth of young speakers has propagated in different ways and to different degrees throughout several generations; different indigenous communities have handled this in varying ways. This has resulted in a necessity to make distinct as well as classify various "stages" of endangerment based upon several factors. The number of speakers of a language is a relatively quick way to establish the severity of a potentially endangered language. A language is considered to be severely endangered when there are less than 25 people speaking the language. The primary age group of speakers sheds more light on the situation. A language is severely endangered when the main/only speakers are no longer virile and may be of failing health, which occurs around the age of 65 in the United States (Ethnologue). When adults that both can and do have children teach those children the language, it is placed on a more secure footing. The largest spoken domain is a way to tell if a language is losing, maintaining, or increasing its base of speakers. Institutionalized use, such as in schools and businesses, generally indicates that a language is maintaining or increasing the number of speakers. Thus, one could be either steady or declining when only spoken in the home, or, more severely in decline when only spoken ceremonially.

One of the issues with classifications being created separately is that even though the diverse categorizations generally agree on concepts of endangerment, some are significantly more granular

than others. For example, Dr. Joshua A. Fishman has defined eight stages of endangerment and below is a chart based on his book, Reversing Language Shift, which outlines his definitions (Fishman, 1991).

Table 1: Stages of Endangerment

Stage	State	
1	Some use in higher government levels and higher education	Institutionalized Use
2	Used by local government and community media	
3	Used in businesses and by employees	
4	Usage required at grade school level	
5	Alive, but used only in the community	
6	Spoken by some children and teens	
7	Only spoken by middle-aged adults and older	
8	Only spoken by elders	

This chart gives clear insight to the different levels of endangerment, and the progression that could ultimately happen if a language is not preserved properly.
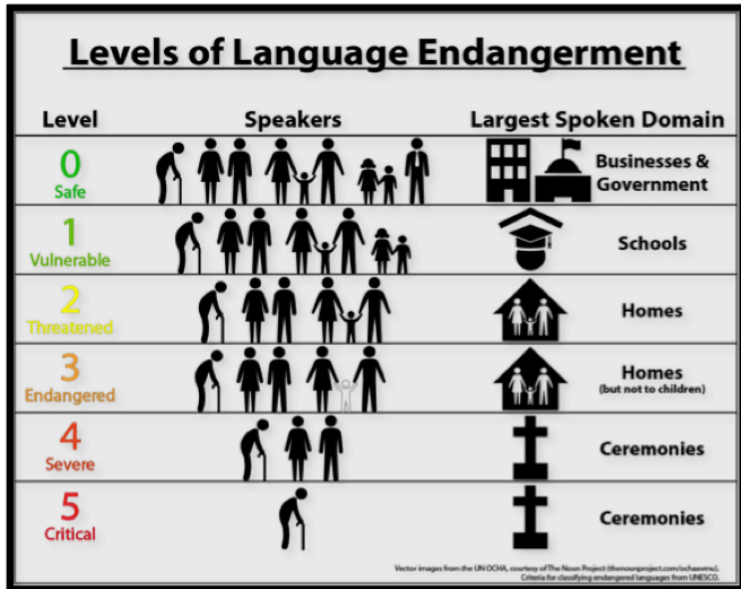
Figure 3: UNESCO Levels of Language Endangerment

Similarly, UNESCO has 6 different levels by which they classify the endangerment of a language. Compared to other resources, UNESCO has significantly more explanations for the kinds of speakers, spoken domain, and number of people speaking the language. The chart below depicts UNESCO's given information.

Because UNESCO's descriptions are incredibly thorough, their classifications are then required to be less rigid; for example, Northern Tujia, has a large enough number of speakers to fall into the "Safe" category, but other factors result in the language still being

considered "Threatened" (Endangered Languages Project). These stages of language endangerment are extremely helpful in determining the best practices regarding language revitalization.

2.2 Endangered Languages in the Santa Fe Area

The southwestern United States is home to the large majority of both Native languages and speakers, with 20% of all speakers residing between McKinley County, NM, and Apache County, AZ, according to the 2010 US Census. There are 19 pueblos with 5 different indigenous languages spoken around the Santa Fe region (Indian Pueblo Organization).
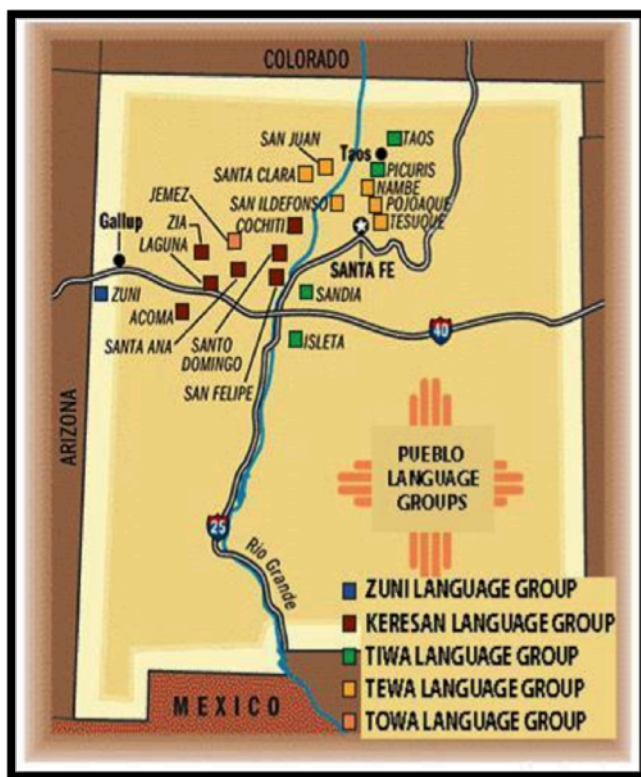
Figure 4: Indigenous Languages in the Santa Fe Area

2.2.1 Keres

The Keres Language has split into two different dialects: Eastern Keres and Western Keres. Due to external pressure, not much history has been compiled about these pueblos that give information about the area and lifestyle before the pueblos became Census-Designated Places. A Census-Designated Place is a concentration of population identified by the United States Census Bureau for statistical purposes. CDPs are outlined for each decennial census as the numerical counterparts of incorporated places such as cities, towns and villages. CDPs have a large enough population to be counted as an area, but lack a separate municipal government ("Census-designated place" 21 February 2013).

Regarding Western Keres, the Laguna Pueblo and the Acoma Pueblo are federally recognized Native American tribes. The Laguna Pueblo has a population of over 7,000 enrolled members, making it the largest Keresan-speaking tribe. The Laguna Pueblo was founded in approximately 1699. The oral history of the Pueblo, however, indicates that they have occupied the area for a much longer period of time. The first contact with the Spanish occurred when a

8

Franciscan friar claimed the Pueblo area for Spain in 1539. By 1616, nine missions had been built in various villages associated with the Laguna Pueblo. Many of the missions were built under the supervision of the Franciscan friar that utilized Laguna labor. These missions were then used to try and convert the pueblo to Christianity. This effect is still seen in the Laguna Pueblo today; many of the traditional ceremonies now have Christian prayers in Keres ("Pueblo of Laguna History"). The history surrounding the Acoma Pueblo is very similar to the Laguna Pueblo; the change, however, came in 1598 when the Acoma tribe retaliated to the Spanish encroachment by attacking a group of Spanish soldiers, killing eleven. In response, the Spanish burned most of the village and killed more than 600 people while imprisoning approximately 500 others. The surviving people were forced into slavery, and men above twenty-five years of age had their right foot amputated. This act displaced the Acoma Pueblo far from their ancestral homeland, and reduced the native speaking population in the pueblo itself (Zax May 1, 2008).

Table 2: Endangered Languages in Santa Fe

Language	Number of speakers in New Mexico	Chief speakers	Status
Tewa	1,500	Older adults	Severely Endangered
Tiwa	1,600	Adults (middle- and old-age)	Definitely Endangered
Keres	4,580	Adults, young adults	Definitely Endangered
Jemez/Towa	3,000	Adults, young adults	Vulnerable
Zuni	9,686	Adults, children	Vulnerable

2.2.2 Tewa

The Tewa language is spoken in six different pueblos in the Santa Fe area. Two of the most well documented pueblos are the Tesuque Pueblo and the Pojoaque Pueblo. The Tesuque Pueblo has been in its current location since the thirteenth century. The Tesuque Pueblo is known for its fierce adherence and reverence to its traditional religious ceremonies. It has stubbornly resisted all efforts by the Spanish and other invaders to change these religious ceremonies. The Tesuque Pueblo is one of the most traditional of the Tewa speaking pueblos, despite having been in contact with outside cultures throughout much of its history ("Tesuque Pueblo").

The Pojoaque Pueblo is another Tewa speaking pueblo around the Santa Fe area. According to archeologists estimate the Pueblo has been inhabited since the early 2 century. Pojoaque has always maintained a strong cultural identity and was known by its Tewa speaking neighbors as "Po-suwae-geh", meaning the water drinking or gathering place. This strong cultural identity was put to the test during the late 17th century when the Spanish pushed the Pojoaque people from their homes into nearby pueblos. It was briefly re- inhabited by five Pojoaque families during the 1800s; in 1936, the pueblo became a federally recognized Tribal Reservation, and has had a strong economic rebirth due to Tribal economic development efforts in the Pojoaque basin area ("Pojoaque Pueblo").

2.2.3 Tiwa

The Tiwa are group of Tanoan pueblo peoples in the Santa Fe region. They traditionally spoke the Tiwa language and were divided into two groups: Northern Tiwa, which is spoken in Taos and Picuris, and Southern Tiwa, which is spoken in Isleta and Sandia ("Tiwa People" 1 March 2013). Most archaeologists believe that the Taos Native Americans settled along the Rio Grande in the late 13th century due to water supply being more reliable. The first Spanish explorers found the Taos Pueblo in 1540. Around 1620, the first Catholic Church in the pueblo, San Geronimo de Taos, was built. The construction of the church was the beginning of the culture clash between the natives and the Spanish. Reports from the period indicate that the native people of Taos resisted the building of the church and the imposition of the Catholic religion. Throughout the 1600s, cultural tensions grew between the native populations of Taos and the increasing Spanish presence. New Mexico formally became a territory of the United States in 1847. The Taos Pueblo's 48,000 acres of mountain land was appropriated by President Theodore Roosevelt and designated as the Carson National Forest in 1908 ("National Forest of the US").

President Nixon returned a portion of the mountain land in 1970, and the rest was transferred back to the Pueblo in 1996. Blue Lake, which the people of the Pueblo traditionally consider sacred, was included in this final return of Taos land. The Taos people label the acquisition of the sacred Blue Lake as the most important event in their history because of the spiritual belief that the Taos people originated from the lake itself ("Public Law 104–333" Nov. 12, 1996) ("Pueblo de Taos") ("Taos Pueblo, Taos New Mexico").

The Sandia Pueblo's culture developed from 700–1100 AD, characterized by its distinctive religious beliefs and practices and a large growth in population. The Sandia Pueblo has been

populated in its current location since the 14th century. Records show that the Spanish conquistador Francisco Vásquez de Coronado came into contact with the Pueblo of Sandia in 1539. In 1617, the Spanish began to exact tributes and enslave members of the Sandia Pueblo for labor in the building of churches and working in Mexican mines. The resentment of the Sandian people against this abuse drove the Spanish from the region until 1692 when Diego de Vargas conquered it. With the end of the Mexican-American War in 1848, the territory of New Mexico was ceded to the United States ("Pueblo of Sandia History").

2.2.4 Towa

The first recorded history from the Jemez people states that they arrived in the Santa Fe area in the late 13th century. In 1541, the Jemez People came into contact with Spanish Conquistadors; this eventually led to the Jemez People being concentrated into a small village, now known as Walatowa, in the early 1700s. In 1838, the Jemez community experienced a large population boost from the Towa speaking people of the Pueblo of Pecos. The Pueblo of Pecos relocated to Walatowa due to many external pressures. The Towa language spoken in Walatowa has remained largely unmodified throughout the course of history due to the many strict rules about its use ("History of The Pueblo of Jemez").

2.2.5 Zuni

The first archeological evidence suggests that the Zuni were present around the Santa Fe area almost 4,000 years ago. The Zuni People have been located at their current position since 1692. Due to raids from the Apache, Navajos, and Plains Indians, the Zuni Pueblo applied for a reservation grant from the United States of America. The Zuni Pueblo became an official Native American Reservation in 1877. Throughout the mid-19th century, the Zuni Pueblo moved away from traditional farming and more towards sheep and cattle herding for economic stability ("Zuni People" 23 February 2013).

2.4 Language Revitalization

There is a strong argument made as to why indigenous languages ought to be preserved. Abutters argue that a single universal language is the direction the world should be heading in, and having multiple languages only inhibits progress to be made in the future. However, in order to make progress, one has to account for the steps that were previously taken to build upon what has

already been established. The significant detriments to losing an exorbitant number of languages in the world include the loss of understanding of their cultural writings and the amount of knowledge contained in the history of their people (Crawford, 1995). The purpose of every language is to communicate between two individuals; over the course of history, these languages develop and contain the experiences of the culture and society that used it. "Many endangered languages have rich oral cultures with stories, songs, and histories passed on to younger generations, but no written forms. With the extinction of a language, an entire culture is lost (National Geographic, 2012). If history is unknown, it is doomed to be repeated. By preserving these languages, we will ensure that today's society will move in a positively linear direction.

2.4.1 Successful Revitalization

As challenging as it might seem to overcome a language that is facing extinction, there have been languages revitalized; a common example is Scottish Gaelic. At the end of the 19th century, there was a sharp decline in the Gaelic language. This was due to the Education Act of 1872, which discouraged the use of Gaelic. Because of this, Gaelic was restricted and the language of English was promoted. As time progressed, recent language policies showed that there had been an increased support of Gaelic. At the turn of the millennium, Scotland held elections to change their government to Scottish Parliament, which had not been done in nearly 300 years. Because the Scottish Parliament oath was partly Gaelic, this boosted the number of speakers. In 2005, a "Bill for an Act of the Scottish Parliament to establish a body having functions exercisable with a view to securing the status of the Gaelic language as an official language of Scotland" was passed ("Gaelic Language (Scotland) Bill" 2012). This act was entitled the Gaelic Language Act. Under this act, an official language development body was created: Bòrd na Gàidhlig. This organization's responsibilities include:

- To give advice to Scottish Ministers on Gaelic matters;
- To prepare and steer the National Plan for Gaelic;
- To give advice on Gaelic and Gaelic education in Scotland;
- To provide guidance to public bodies in Scotland on Gaelic language plans; and
- To monitor the implementation of the European Charter for Regional or Minority Languages, and report the findings to the Scottish Ministers.

("Gaelic Language (Scotland) Bill" 2012).

Because of these changes that were required, there was a substantial growth in fluent speakers. The language was fully revitalized and Scotland is now a bilingual country. Scotland took the approach of incorporating the programs into their government to make them a regulation of the country; while this is not feasible in every case, taking initiative with some of these programs will surely help any language seeking revitalization.

3. Methodology

We have labored to produce our goals of working with the Indigenous Language Institute to provide accessible, up-to-date resources that will help revitalize endangered indigenous languages. We achieved this by completing three important objectives:

- Assisting the Indigenous Language Institute in updating the Native Languages Revitalization Resource Directory.
- Providing an online site that offers access to the resource directory in an intuitive manner
- Maintaining the online resource directory by easing the expanding and updating processes[1]

Our primary objective was to update the Indigenous Language Institute's current resource directory. The Native Languages Revitalization Resource Directory contains organizations, individuals, and private funders that work towards preserving languages and culture. The directory also contains a bibliography of relevant books and articles. This directory was developed in 1997 and was published in 1998, and has been updated sparingly. As such, we updated the directory to reflect the newer ways to collect, organize, and share information. Our second objective was to create an intuitive user interface, which permits someone on the web to access the updated directory. The current directory is very unhelpful when trying to find information when compared to the recent strides made in ease of access technology. By adding this interface it allowed the information contained in the directory to be shared with many more people around the globe. The last goal that we have accomplished is to ensure the maintenance of the directory. To do this, we have made updating and expanding the directory as simple as possible. Additionally, we created and published training materials for our sponsors that allows them as the administrator to update and add to the directory after we leave Santa Fe and the Indigenous Language Institute.

---

[1] Please note that the third objective was accomplished after the creation of the online resource directory, so please refer to *4.2 Maintaining the Online Resource Directory* for more information.

3.1 Updating the Resource Directory

The resource directory that was produced and published by IPOLA in 1998 was used as a source of information for those looking to revitalize their native tongue. As language revitalization started moving forward and modern technology became more commonplace, the paper-based directory fell by the wayside. As such, the entries in the directory are now significantly out of date. Sometime later the ILI, in an effort to bring the IPOLA's directory to modern technology, put a fairly similar version of the resource directory online in a subpage of the Indigenous Language Institute website in 2000; however, when the version was posted online, it was barely updated. Having a resource directory nearly thirteen years out of date, our first goal was to update the directory and compile it in such a way that it will be updated much more regularly than it was in the past. To accomplish this, we compiled the information we already have in both forms of the directory and began to update from there.

3.1.1 Converting the Online Program Directory

The online version of the ILI's Program directory is stored in a document format known as Extended Markup Language (XML) where each entry and type of data has its own tag, arranged hierarchically, and each entry in the Directory has its data tagged accordingly. For the use of this data by our group, we created a script that pulled the XML-formatted data from the online Program Directory and compiled it into a simple, spreadsheet-styled format by the name of Comma-Separated Value, or CSV. We chose Python, a dynamic programming language, to do this because Python is noted for its clear, readable syntax and its high-level dynamic data types (Python Programming Language, 2012). The spreadsheet that was generated from the Python program we wrote was imported into a Google Spreadsheet so that we could collaboratively edit entries and track all of the changes made to it.

3.1.2 Digitizing the IPOLA Paper Resource Directory

The paper-based Resource Directory required significantly more legwork to create usable data in a process known as "scraping". The pages of the directory have been scanned to images; those images were run through an Optical Character Recognition (OCR) program to convert the image of each page into a text file. Due to the more irregular formatting of the paper Resource Directory, processing these files with a program required too great of an investment in programming time. Duplicates also existed between the online Program Directory and Paper-based Resource

Directory, and significant additional programming was required to cross-reference the two sources. Due to these nuances, we instead catalogued the contents of the paper-based directory by hand, manually appending the entries that were otherwise missing into the spreadsheet.

In addition to the data we have acquired from these sources, we have added columns to explicitly state various extra fields of data such as the type of information, language family, etc. These additions were for when the contents of this spreadsheet were used to create online entries, which have additional tags about the data, commonly known as "metadata", that enable richer web searches and digital cataloging. This compiled spreadsheet served as the crux of the rest of the project, built upon in the next steps. Please see Appendix B for the compiled spreadsheet template.

3.1.3 Integrating and Validating the Resource Directory

The next step that was taken for the updating process of the resource directory was manually going through each of the directory entries located in the spreadsheet and updating the contact information. We did this through a variety of ways. First, if the entry included it, we visited the programs' websites with the links provided. Many of these websites had cumbersome layouts, making the information we were seeking difficult to find. If the directory entry did not have a website available, we used a search engine to gather the location data if it had changed. There were a small number of associations that we could find absolutely no information for. We called the telephone number listed in the old directory, however, in each case the phone number was disconnected. We labeled these affiliations as inactive. Additionally, in a few cases, the name of the organization or program had changed, creating an additional step in our search for the appropriate material. With all of the new information we gathered, we entered that into our existing Google Spreadsheet, combining it with the previously generated information.

Figure 5: Resource Questionnaire

Another means to updating the resource directory is an online questionnaire that was sent out to each of the language programs in our updated Google Spreadsheet (please refere to Appendix C). This questionnaire was broken up into three sections. The first part is complete with fields for the language programs to update their contact information and elaborate on their program information. There is a second section in the questionnaire for organizations associated with indigenous language programs or funding organizations/sources. The last section in the questionnaire is a field where the individuals can inform us of any additional language programs that would like to be included in the updated resource directory; this is due to the likelihood of language programs existing that do not have a strong web presence. With these added language programs we receive, we will contact them directly using the information provided by the surveyor.

3.1.3 Categorizing The Resource Directory

To create a more organized resource directory, we have decided to differentiate between the various types of resources that are available for language revitalization. After speaking with our sponsor, we established three different types of resources: Organizations, Programs, and Funding Sources. Organizations coordinate different events to gather those involved in language revitalization. Language Programs are labeled as such because these facilities offer a multitude of programs that help directly with language revitalization, such as classes, seminars, and workshops. Funding Sources are a special type of resource in the sense that while they do not themselves enhance language revitalization, they provide financial support to organizations and programs that do, which allows the language programs to flourish given the additional means of sustenance. With these three divisions of resources, the directory will have significantly more coordination and structure, all of which will add to the search-ability of the new research directory.

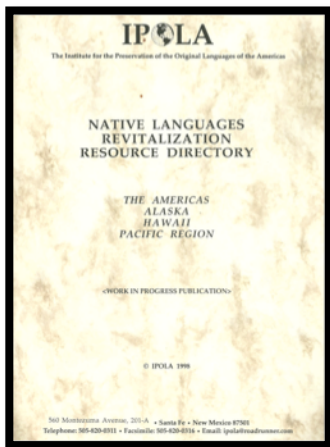## 3.2 Creating a Wiki Version of the Resource Directory

Figure 6: Paper-based Directory

Without making our efforts publicly available, they are all for naught. This is why we published the contents of this spreadsheet onto the Web. The full Native Languages Revitalization Resource Directory is currently a paper-based document. It is categorized into groups such as language stabilization programs and program sponsors, and entries are listed alphabetically by name or organization. This makes searching for resources by any other means, such as location or language, extremely inefficient. In addition to this reduced functionality, the directory has limited access due to its hard-copy design.

The present online content the ILI possesses is a nearly-plaintext list of programs, which is unpublicized and located in a section of the ILI's website that is not easily navigable. It is formatted in a similar manner to the paper-based version, where entries are alphabetically listed by program name. It is currently lacking in features that regular Internet users seek, such as search functionality and the ability to click on the links and email addresses on the page to make use of them.

Figure 7: Online Program Directory

3.2.1 Making Model Wiki Pages

To develop a general sense of what these pages will look like, we created model pages on the Wiki Sandbox, which is a tool used by Wikipedia that is a temporary area to test web page creations. These model pages are actual Language Programs, Organizations, and Funding Sources (Appendix D). We used the Pueblo of Isleta for the model Language Program, the Indigenous Language Institute for the Organization, and the Lannan Foundation for the model Funding Source. We have made these pages look as professional and detailed as needed while still maintaining the spirit of Wikipedia pages. The thought process behind these model pages was to exemplify what an ideal page would contain for information, layout style, and organization.

To match the functionality we became accustomed to in the Wiki Sandbox, we installed the following MediaWiki extensions:

Table 3: MediaWiki Extensions

Cite	Adds Wikipedia-style citations
Maps	Adds the ability to embed customized Google Maps and the geocoding of addresses and location data
ParserFunctions	Adds logical functions to the Wiki syntax parser (if, ifeq, switch, etc.)
SemanticMediaWiki	Adds metadata tagging and page queries
SemanticCompoundQueries	Adds more sophisticated page querying abilities
SemanticMaps	Adds Semantic query functionality to the Maps extension
Validator	Basic support for other extensions
Vector	Enhancements for the default "Vector" MediaWiki skin
WikiEditor	Provides the editing interface

These extensions give us the ability to style and customize the layout of the Directory entries to a greater degree. They add richer media to enhance the design of the entry pages, and allow us to increase usability.

3.2.2 Gathering Feedback

When updating this resource directory, something that is of great prominence is the opinion of whom it will directly affect: the resources themselves. We selected the aforementioned model pages because we had the privilege of meeting with each of the different affiliations. We showed them their respective pages and we asked for the opinions on the site layout and content. Additionally, we allowed them to practice editing their page, and thus acquired feedback on what they would need assistance on as far as editing and updating their site. Based on the input we received, we made the appropriate changes to the entries; for example, we adjusted the names of the headings and placement of information. We observed and documented people editing the pages, and took note on what specific actions needed clarification.

3.2.3 Publishing the Wiki Pages

After creating the templates to be used for the different resources and updating each of the directory entries, we combined these and generated the pages for the online directory. We have done so by creating a script in the Python programming language that processes the spreadsheet and creates the pages based on the styling of the model pages. The pages by themselves are only useful when they are placed on the Resource Directory for public access and use.

Plug-ins exist to connect Wiki entries and spreadsheets or other databases, but these require that the pages remain connected, and any changes to the content of the pages must be made to the database that lies behind the scenes. We chose a more transparent path by only using the spreadsheet in the initial creation of the pages and discarding it after the pages were generated; this resulted in each page containing its own content. We expect this simpler system to be easier to maintain as a result, as there is one less layer of software in the mix.

So that this online directory has an opportunity to expand its user base, the pages also have external links to related programs. Each page has a link to search the directory for a program's associated language, UNESCO's Endangered Language Programme home page, as well as a link that searches for the language on the Endangered Languages Project.

4. The Revived Revitalization Resource Directory
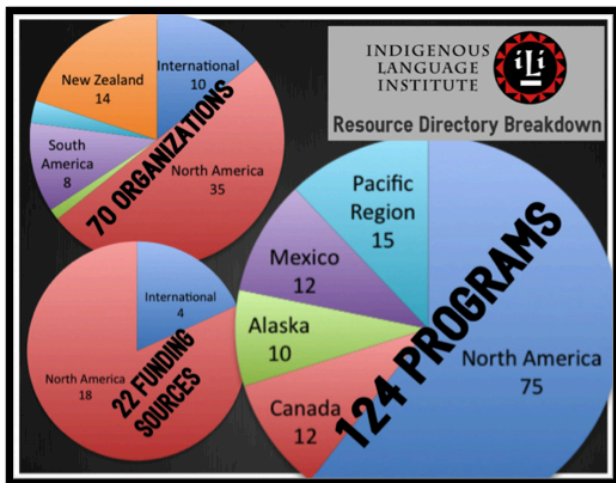
Figure 8: Resource Directory Entries

Our final creation is the Revived Revitalization Resource Directory; this directory is a website based off of Wikipedia, the popular online encyclopedia. It contains pages for the Organizations, Funding Sources, and Language Programs in North America. We have updated 137 programs, 37 organizations, and 21 funding sources that were located in the paper-based IPOLA Native Languages Revitalization Resource Directory. Although we started out with 124 programs, when conducting research we found additional programs; our original number of 124 expanded to 195 programs.

4.1 Overview of Wiki Directory

Our online resource directory has three main enhancements from the paper-based version: accessibility, searchability, and perpetuity. We made these enhancements as significant as possible to ensure that the new and improved version fit in with the modern concept of Internet use, as well as having a profound impact on communication between the language revitalization community. A noteworthy amount of consideration and evaluation went into the design of this directory, and both its attributes and structure reflect the quality of the directory.

Layout
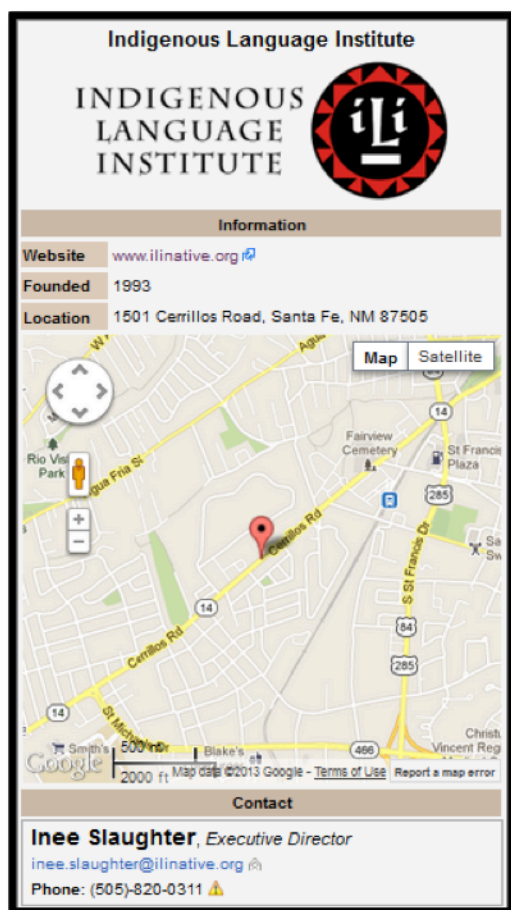
Figure 9: Information Box
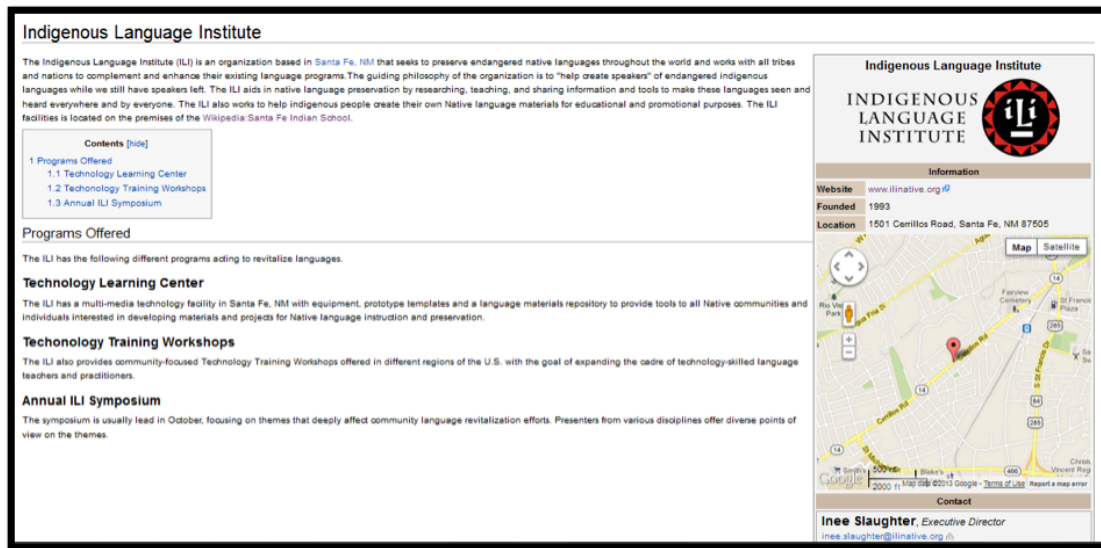
There are a total of 174 entries in the Native Language Revitalization Directory. All 174 pages have a similar formatting style: to the right, there is an Information Box containing basic facts about the program. In each Infobox, there is an interactive map as well as contact information. All of the figures in the Infobox are pertinent information; however, it is pulled to the side as a quick reference.

The beginning of the body section is a brief description about the mission and directive of the organization, funding source, or language program. It is not intended to be a description about the group, nor should it include information relating to administrative topics of the described program.

The contents of the page should be reserved solely for the seminars, conferences, classes, programs, funded programs and grantees. Following each different section there should be a brief description to explain how that particular entry helps revitalize native languages.

Organizations

　　Much like all of the Wiki pages on the directory, the organization page is formatted to have an Information Box on the right of the page, an introduction at the top of the page, and the organization's mission statement and a short description of what they do. The Infobox for the organizations contain the basic information about the organization, such as when it was founded, its location, and a link to the organization's website. Also displayed in the Infobox is the contact information for the liaison with that particular association.

Figure 10: Organization Entry

Funding Sources

　　Funding sources are different from organizations by way of the page contents; in a funding sources content section, listed are the different programs they fund, as well as the different grants they provide. If the funding sources are comfortable with it, they may put the monetary values of the grants and provide a link to their grant application.

Figure 11: Funding Source Entry

Language Programs

The Language Program Infobox contains the language that the program teaches as well as the ISO 639-3 code that is specific to each language. In addition, the content portion of the programs should contain all of the classes, events, and after-school programs offered by the association. These should be described in as much detail as possible. To reduce the need to excessively update the information, however, they should not include information that changes rapidly: for example, the names of the teachers or the class times of various courses. That type of information should be left on the website owned and operated by the specific language program.

Figure 12: Language Program Entry

22

4.1.1 Accessibility

This new online resource directory is located suitably right off the currently existing Indigenous Language Institute website at http://www.ilinative.org/dicrectory. Given the fact that the directory is now online, it will reach a notably larger audience than previously; anyone with Internet access has the capability to reach the directory. This is an extraordinary development for the ILI directory because individuals seeking language revitalization methods are now unimpeded with the difficulty of obtaining the paper-based resource directory.

4.1.2 Searchability
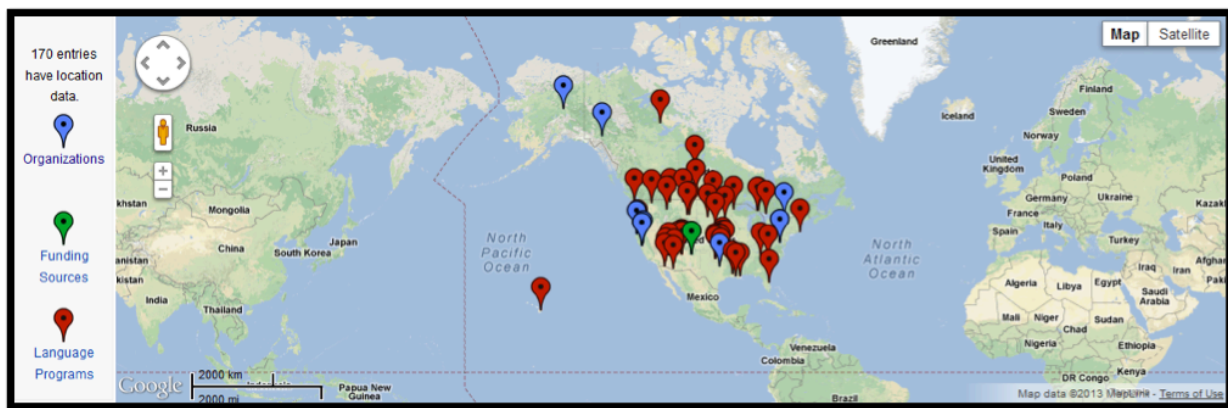
Figure 13: Online Directory Interactive Map

With this online directory, the means for searching have been remarkably strengthened by features such as interactive maps, categories, and a comprehensive search bar. When first accessing the Revitalization Resource Directory, one will be directed to the home page. On this homepage, there is an interactive map that displays the three categories of pages: Organizations, Funding Sources, and Language Programs. These divisions make it easier to find the information when searching.

The interactive map behaves like a normal Google map; it is clickable with the ability to pan in all directions, as well as zoom in and out. These features allow one to find various programs in their vicinity. Once an organization, language program, or funding source is found, clicking on the appropriate pin redirects to the respective entry page. Another way to search for the program or organization is to click on the links located at the top of the homepage. These will open an alphabetized listing of the pages under the selected category.
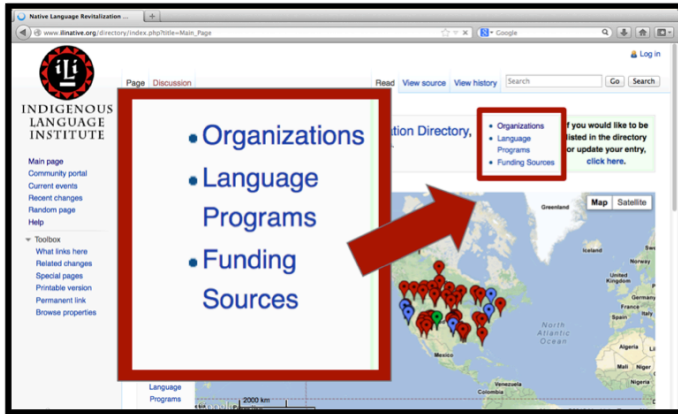
Figure 14: Directory Categories

Further, similar to Wikipedia, there is an extensive search bar at the top of the page that will allow the user to specify precisely what they are questing. If the searcher knows the exact name of the affiliation they are looking for, pressing "Go" will direct them straight to the entry's site. If not, a search query will open with all of the entries containing the search.

4.1.3 Perpetuity

The most innovative feature of this online resource directory is its ability to be continually updated. Formerly, the paper-based directory was only published once; if any changes needed to be made or a program needed to be added, a completely new directory would have to be printed. However, in order to verify that revisions and add-ons will be produced, we analyzed and determined the best possible ways to make certain this directory stays current.

4.2 Maintaining The Online Resource Directory

We collaborated with our sponsors and determined that the Indigenous Language Institute will be the authoritative power in the updating and expanding of the directory because of the ILI's ability to have staff on-hand equipped with a comprehensive knowledge of Wiki markup. We hope this will encourage the owners of the pages to keep the ILI informed of changes in their program information; thus the ILI will refresh the pages with new content at the behest of the organization. This allows for more extensive descriptions of their content than in the previous version while also removing the need to manage a multitude of users.

4.2.1 Updating and Expanding the Directory
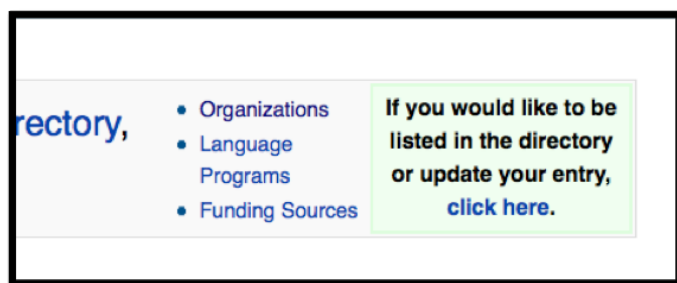
Figure 15: Additions and Updates Link

After creating this online resource directory, we want to be confident that the entries are updated on a regular basis. Moreover, we wish to encourage new resources to contact the ILI to have their program included as well. To do this, we have provided a link on the directory's home page. This link is a button that brings you to an article wizard that we made specifically for our needs.

4.2.2 Establishing Editorial Guidelines

While we want the organizations, groups, and individuals listed in the online database to be able to add as much information as they can, we want to structure the content of the information that is being contributed. For example, it would be inappropriate to add information relating to the archiving of indigenous languages in the database. Additionally, if it were permitted to use the directory as a utility to teach people the languages, it would become another competing avenue for language revitalization; this would be counter-intuitive to the goal of our project. The way we have prevented this from happening is by having an updating/adding entry wizard. A wizard is simply an online series of questions in a flow-chart format that direct the operator to the appropriate area based on your previous answer.

The first page of our wizard is simply an introduction page that outlines what steps need to be taken to complete the form. It displays a welcome message along with directions on filling out the wizard.
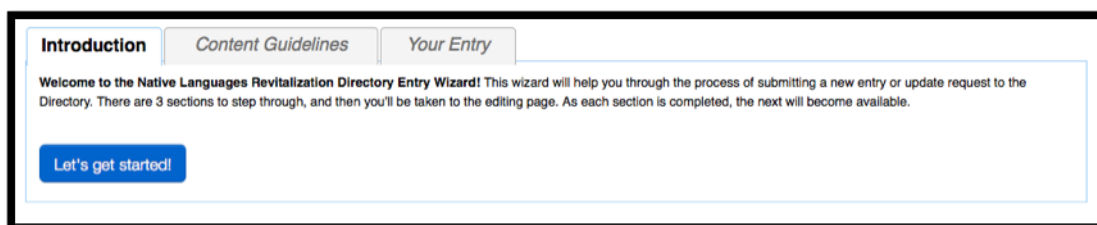
Figure 16: Wizard Introduction

The next section is entitled "Content Guidelines". This is where the directory is explicitly summarized and includes what the directory should include and what it should not include. From there, three options present itself: updating an entry, adding a new entry, or changing the name of an entry.

After a certain option is selected, the third section is entitled "Your Entry". Based on which option was chosen, a different type of window will appear. In each of the cases, a form is shown to be filled out by the user. This form is very similar to the online questionnaire that was used earlier in our project to find out basic information about a particular resource. With this form, one can fill out the appropriate contact information and program information on their association.

4.2.3 Consistency and Ease of Editing

To ensure that this online directory stays congruous and make editing a simpler task, we have created and added Wiki Templates. This was another critical stepping-stone between updating the information and putting it online.

Much of this work was taken care of by the handful of freely available and widely used Wiki templates (eg: Infobox, Navbox, Ambox) and plethora of examples on editing and maintaining them. They served the purpose of negating the need to write raw HTML (even if it's copying and pasting) for each and every styled element on the page.

To adapt these templates for our purposes, we created meta-templates, another common MediaWiki practice. Meta-templates specialize the fields of these generalized templates and pre-populate some of the content to make editing pages a simpler and more consistent task. By assigning meaningful names for the various fields in the markup of the box, a template can be simplified thusly:

```
{{Infobox
|bodystyle= width:30em;
|above = Indigenous Language
Institute
|image = [[File:ILI Logo -
big.png|300px|alt=Logo of Indigenous
Language Institute]]

|headerstyle  = background:#cba;
|labelstyle   = background:#dcb;
|header1  = Information
|label2    = Website
|data2 =
{{URL|ilinative.org|ilinative.org }}[[Has
website::ilinative.org| ]]
|label3   = Founded
|data3 = 1993 [[Was founded::1993|
]]
|label8   = Location
|data8 = [[Has address::1501
Cerrillos Road, Santa Fe, NM 87505]][[Has
coordinates::{{#geocode:1501 Cerrillos
Road, Santa Fe, NM
87505|service=google}}| ]]
|data9 = {{#display_points: <1501
Cerrillos Road, Santa Fe, NM 87505>}}
|header20= Contact
|data21  = {{Contact
|name   = Inee Slaughter
|title  = Executive Director
|child  = yes
|image  =
|email  =
inee.slaughter@ilinative.org
|phone  = (505) 820-0311
|fax    =
}}
[[Has contact::Inee Slaughter| ]]
[[Has phone::(505) 820-0311| ]]
[[Has
email::inee.slaughter@ilinative.org| ]]}}
}}
```

```
{{Infobox Organization
|name       = Indigenous  Language
Institute
|website = ilinative.org

|logo    = ILI Logo - big.png
|location= 1501  Cerrillos  Road,
Santa Fe, NM 87505

|founded = 1993

|cname   = Inee Slaughter
|cimage  =
|ctitle  = Executive Director
|cemail                          =
inee.slaughter@ilinative.org
|cphone  = (505) 820-0311
|cfax    =
}}
```

These two examples produce exactly the same Infoboxes, but, as shown above, every "label"/"data" pair has been replaced with a meaningful name, and every instance of duplicated information has been eliminated. This is accomplished through the use of a handful of powerful MediaWiki parser functions, such that each named value can be used as many times as needed in a template by invoking {{{name}}} to place its value or perform logical analysis to determine what fields to display. In fact, the Infobox on the left would become even more complicated and cumbersome to maintain if the Contact template had not been created, as it was written from scratch for this site.

4.2.4 Writing Text Tutorials

To make the actual editing of the directory entries as easy as possible for the administrator, we created text-based tutorials. This are easily found off the homepage of the directory. Text tutorials will allow users to use commonplace computer hotkeys, such as "copy" and "paste". The

use of these shortcuts will speed up the process of editing the Wiki pages. In the text version, we will provide a list of commonly used Wikipedia syntax for actions including adding a link or embedding a picture.

Figure 19: Wiki Help Cheatsheet

4.2.5 Producing Online Video Tutorials

Camtasia is screen-recording software that enables a user to video record their monitor to capture action that takes place on a computer screen. Another feature of Camtasia is the ability to use common video-editing tools of those recordings. We used Camtasia to produce a step-by-step procedure of Wiki's editing options to guide the user to import, edit, or add the information appropriate to the directory database. We made full use of the Camtasia features and added graphics and callouts in the recording to make sure the point of the tutorial videos are properly conveyed.
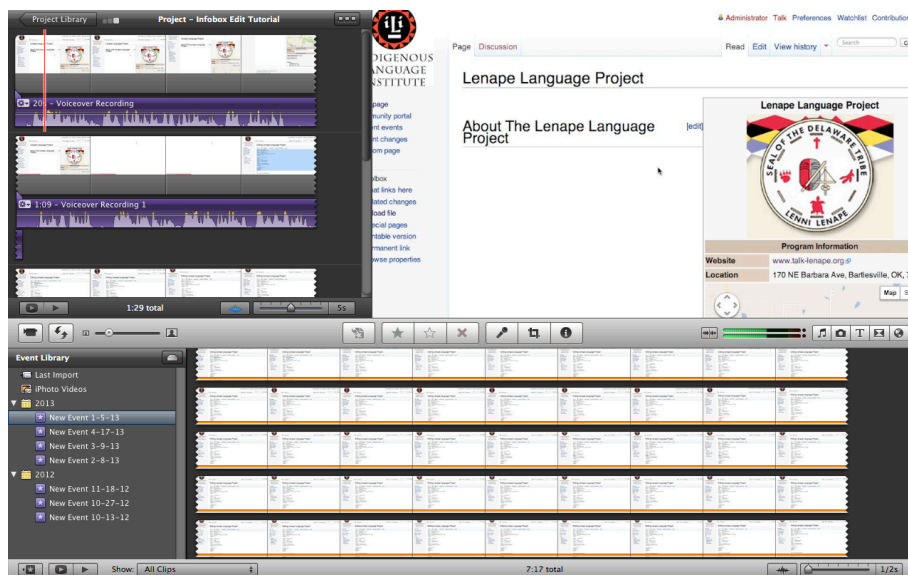
Figure 20: Video Tutorials

5. Recommendations

Our recommendations consist of the other objectives that we would have completed if we had more time; these objectives work to enhance and elaborate upon the directory. We would include programs and functions to keep track of how people use the directory, as well as add another section to include books that are about language acquisition and language learning. Eventually, we want the directory to expand to include programs organization and funding sources outside of North America.

5.1 Improving and Expanding the Directory

These recommendations are devices and features that we feel would be very beneficial to the overall effectiveness of the Resource Directory. These improvements would be to use basic traffic monitoring, expanding the categories listed in the directory to include books and other publications relating to language revitalization, opening the directory to include programs from around the globe, and, finally, we would hope that changes can be made to the MediaWiki editing process.

In order to improve the quality of the resource directory, we think that a system providing information on how people use the directory would be advantageous. This data would allow us to see what links and features are used the most, as well as, more importantly, what features are not being used. We would use the data to improve the functionality of the web-based tools on the directory.

Adding a Bibliography

The goal of this directory is to be the complete source of revitalizing languages through programs and classes. We hope to eventually include a list of the books that would allow those working towards language revitalization to read about the various books and documents relating to the subject.

Including Programs Outside of North America

Another recommendation for the improvement of the resource directory is to continue to expand the number of resources offered in it. A simple way to do this is to work with larger

organizations, such as the Endangered Languages Project or UNESCO, to see if they could provide information about the various language programs that they work with. Another way to expand the directory is to discover and contact new global funding sources and include the programs that they fund.

5.2 Publicizing and Sharing the Directory

Making the directory well known is a major component to the overall success of it. In order to make the directory become a widely accepted and respected device used to revitalize languages, there needs to be seminars and classes about the directory as well as sheer awareness on many different social media facets.

Hosting Seminars Centered on the Directory

In order to increase the legitimacy of the directory, as well as improve the report with the indigenous communities, the ILI should bring the paper-based questionnaire to all of the conferences and seminars that the ILI attends. Another beneficial event that the ILI could sponsor is a convention dedicated to the improvement of the Resource Directory.

Incorporating Social Media Into The Directory

As the directory continues to grow, social media is another source of publicity. Creating a Facebook page for the ILI and the directory will be a favorable way to reach out to the younger generation of native speakers. As more young native speakers begin to use the directory to find ways to revitalize their language, they will bring new and refreshing ideas to add to improve language revitalization.

6. Works Cited

Abrams, D. M. 2003. Modelling the dynamics of language death. Nature (London) 424 (2):900.
    Crawford, J. 1995. Endangered Native American languages: What is to be done, and why?
    Bilingual research journal 19 (1):17.

Edinburgh, Scotland: 2012. s.v. "Gaelic Language (Scotland) Bill."
    http://www.scottish.parliament.uk/parliamentarybusiness/Bills/25174.aspx (accessed).

Fishman, Joshua A. 1991. Reversing language shift: Theoretical and empirical foundations of
    assistance to threatened languages. Vol. 76: Multilingual Matters Limited. Forest History,
    "National Forest of the US." Accessed March 2, 2013.
    http://www.foresthistory.org/ASPNET/Places/National Forests of the U.S.pdf.

Hale, K. 1992. Endangered languages: on endangered languages and the safeguarding of diversity.
    Language (Baltimore) 68 (1):1.

Hinton, Leanne. 2008. Vol. Bingley UK: Emerald.

Indian Pueblo Cultural Center, "Pojoaque Pueblo." Accessed March 2, 2013.
    http://www.indianpueblo.org/19pueblos/pojoaque.html.

Indian Pueblo Cultural Center, "Tesuque Pueblo." Accessed March 2, 2013.
    http://www.indianpueblo.org/19pueblos/tesuque.html.

Jones, Mari C. 2013. Language Obsolescence and Revitalization.

McCarty, Teresa L. 2003. Revitalising Indigenous Languages in Homogenising Times. Comparative
    education 39(2):147-163.

National Geographic, "Pueblo de Taos." Accessed March 2, 2013.
    http://travel.nationalgeographic.com/travel/world-heritage/pueblo-de-taos/.

National Geographic. (2012). Retrieved from
    http://travel.nationalgeographic.com/travel/enduring-voices/ National Park Service, "Taos
    Pueblo, Taos New Mexico." Accessed March 2, 2013.
    http://www.nps.gov/history/nr/travel/american_latino_heritage/Taos_Pueblo.html

Nationalencyklopedin "Världens 100 största språk 2007" The World's 100 Largest Languages in
    2007.

Public Resource, "Public Law 104–333." Last modified Nov. 12, 1996. Accessed March 2, 2013.
    https://bulk.resource.org/gpo.gov/laws/104/publ333.104.pdf.

Pueblo of Laguna, "Pueblo of Laguna History." Accessed March 2, 2013.

http://www.lagunapueblo-nsn.gov/test2.aspx.

Pueblo of Sandia, "Pueblo of Sandia History." Accessed March 2, 2013.

http://web.archive.org/web/20080102031937/http://www.sandiapueblo.nsn.us/history.ht
ml.

Reyhner, Jon, Gina Cantoni, Robert N. St. Clair, and Evangeline Parsons Yazzie. 1999. Revitalizing
Indigenous Languages. Flagstaff, Arizona: Northern Arizona University.

Romaine, Suzanne, and University of Oxford Merton College. Preserving Endangered Languages.
Language and Linguistics Compass 1 (1-2):115-132.

Walatowa Visitor Center, "History of The Pueblo of Jemez." Accessed March 2, 2013.

http://www.jemezpueblo.com/content.asp?CustComKey=364865&CategoryKey=364886&
pn=Page&DomName=jemezpueblo.com.

Wikipedia, "Acoma Pueblo." Last modified 25 February 2013. Accessed March 2, 2013.

http://en.wikipedia.org/wiki/Acoma_Pueblo

Wikipedia, "Census-designated place." Last modified 21 February 2013. Accessed March 2, 2013.

http://en.wikipedia.org/wiki/Census-designated_place.

Wikipedia, "Tiwa People." Last modified 1 March 2013. Accessed March 2, 2013.

http://en.wikipedia.org/wiki/Tiwa_people.

Wikipedia, "Zuni People." Last modified 23 February 2013. Accessed March 2, 2013.

http://en.wikipedia.org/wiki/Zuni_people.

Woodbury, A. C. 2012. What is an Endangered Language? First language 88 (3).

Zax, . Smithsonian, "Ancient Citadel." Last modified May 1,2008. Accessed March 2, 2013.

http://www.smithsonianmag.com/travel/da-ancient-citadel.html.

Appendices

Appendix A: Online Directory Conversion Script - ILI_directory_scraper.py

```
# XML to CSV converter
# Written for WPI SF13-ILI IQP
# Kevin Janesch
# Susan Stukas
# Will Grebe


# usage:
# bash$ python ILI_directory_scraper.py


from xml.dom.minidom import parse
import csv


#this saves the trouble of copy-pasting a LOT of code
def getElement(entry, tag):
        try:
        return entry.getElementsByTagName(tag)[0].childNodes[0].nodeValue
        except:
        return u''


#program start
#XML Tags
xtag_entry = "programEntry"
xtag_program = "program"
xtag_description = "programDescription"


xtag_phone   = "phoneNumber"
xtag_fax        = "faxNumber"
xtag_address = "address"
xtag_website = "webSiteAddress"
xtag_contactemail    = "eMailAddress"
xtag_contactName = "contactName"
xtag_contactPosition = "contactPosition"
#inpath = "http://www.ilinative.org/share/programDirectory/programDirectory.xml"


#files
inpath = "programDirectory.xml"
outpath = "scraped_programDirectory.csv"


#open the XML and CSV files
dom   = parse(inpath)
sheet = csv.writer(open(outpath, 'wb'), delimiter=',')


#write column headers
sheet.writerow([xtag_program, xtag_description, xtag_phone, xtag_fax, xtag_address, \
                xtag_website, xtag_contactemail, xtag_contactName, xtag_contactPosition])

#go through all the entries and make them into spreadsheet rows
for entry in dom.getElementsByTagName(xtag_entry):
        sheet.writerow([getElement(entry, xtag_program),\
                        getElement(entry, xtag_description),\
                        getElement(entry, xtag_phone),\
                        getElement(entry, xtag_fax),\
                        getElement(entry, xtag_address),\
                        getElement(entry, xtag_website),\
                        getElement(entry, xtag_contactemail),\
                        getElement(entry, xtag_contactName),\
                        getElement(entry, xtag_contactPosition)])


#end
```

Appendix B: Directory Templates

Templates `Infobox`, `Ambox`, and `Navbox` are used with minimal alteration from Wikipedia's WikiProject Transwiki (http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Transwiki), which provides a number of basic, portable templates for beginning MediaWiki sites.

Custom Templates

These templates and metatemplates have been written from scratch for the purposes of use on the Directory.

Template:Contact

```
{{#if:{{{name|}}}|<table                 style="text-align:left;outline:1px          solid
#aaa;background:transparent;{{#ifeq:{{{child|}}}|yes|width:100%|width:35em}};"><tr> <!-- Image --
>              {{#if:{{{image|}}}|<td             style="align:center;padding-left:3px;padding-
right:6px;width:{{#if:{{{imagesize|}}}|{{{imagesize}}}|100px}};">[[File:{{{image}}}|border|top|{{
#if:{{{imagesize|}}}|{{{imagesize}}}|100px}}]]</td>}}<td       style="padding:1px      0px      0px
5px;{{#if:{{{image|}}}|border-left:1px     solid     #ccc;}}"><span     style="font-size:100%;"><span
style="font-size:135%;">'''{{{name}}}'''</span>{{#if:{{{title|}}}|<span style="font-size:110%;">,
'''{{{title}}}'''</span>}}        [[Has        contact::{{{name}}}|        ]]        {{#if:{{{email|}}}
|<br/>[mailto:{{{email}}}                      {{{email}}}]                      [[Has
email::{{{email}}}|             ]]                     }}{{#if:{{{site|}}}|<br/>{{URL|{{{site}}}|Contact
Form}}         }}{{#if:{{{phone|}}}         |<br/>'''Phone:'''          {{{phone}}}          [[Has
phone::{{{phone}}}|       ]]       }}{{#if:{{{fax|}}}|<br/>'''Fax:'''       {{{fax}}}       [[Has
fax::{{{fax}}}|  ]]  }}{{#if:{{{address1|}}}|<br/>'''Address:'''<br/><span  style="border-left:1px
solid                                        #ddd;padding-left:5px;margin-
left:30px;">{{{address1}}}</span>{{#if:{{{address2|}}}|<br/><span  style="border-left:1px  solid
#ddd;padding-left:5px;margin-left:30px;">{{{address2}}}</span>  {{#if:{{{address3|}}}|<br/><span
style="border-left:1px                    solid                    #ddd;padding-left:5px;margin-
left:30px;">{{{address3}}}</span> }} }} }} </span></td></tr></table>}}<noinclude>
This is a template that makes presenting contact information much easier and nicer-looking.
<!--made from scratch by the students of the Santa Fe '13-ILI IQP group, Spring 2013: Kevin
Janesch, Susan Stukas, and Will Grebe-->
==Usage==
<pre>{{Contact
 |name   =
 |title  =


 |image  =
 |site =
 |email  =
 |phone  =
 |fax    =
 |address1 =
 |address2 =
 |address3 =
<!--advanced options-->
 |imagesize =
 |child =
}}</pre>


'''imagesize''' defaults to ''100px''. The image size can be set to anything, but may look
strange.


'''child''' defaults to ''no''. Set it to ''yes'' to set the width of the box to 100% of the
element in which it's contained. '''Only''' do this when embedding in other templates, like
Infoboxes.
==Example==
<table>
```

```
<tr><td width=50%>
<pre>
{{Contact
 |name   = Firstname Lastname
 |title  = Occupation Title


 |image  = file.image
 |site = example.com
 |email  = email@example.com
 |phone  = 555-555-5555
 |fax    = 555-555-5556
 |address1 = address1
 |address2 = address2
 |address3 = address3


 |child=no
}}
</pre>
</td>
<td width=50%>{{Contact
 |name   = Firstname Lastname
 |title  = Occupation Title


 |image  = file.img
 |site = example.com
 |email  = email@example.com
 |phone  = 555-555-5555
 |fax    = 555-555-5556
 |address1 = address1
 |address2 = address2
 |address3 = address3
}}</td></tr>
</table>
</noinclude>
```

Template:Infobox Organization

```
{{#if:{{{name|}}}|<!--
SEMANTIC MEDIAWIKI TAGS
-->{{#if:{{{city|}}}} | [[In city::{{{city}}}| ]] }}<!--
INFOBOX
-->{{Infobox
|bodystyle= width:30em;{{{bodystyle|}}};
|above  = {{{name}}}
|image = {{#if:{{{logo|}}}|[[File:{{{logo}}}|300px|alt=Logo of {{{name}}}]] }}


|headerstyle  = background:#cba;{{{headerstyle}}};
|labelstyle   = background:#dcb;{{{labelstyle}}};
|header1  = Information
|label2   = Website
|data2 =        {{#if:{{{website|}}}|       {{URL|{{{website}}}|{{{website}}}       }}       [[Has
website::{{{website}}}| ]] }}
|label3   = Founded
|data3 = {{#if:{{{founded|}}}| {{{founded|}}}[[Was founded::{{{founded|}}}| ]]}}
|label8   = {{#if:{{{location|}}}| Location}}
|data8 = {{Location|{{{location|}}} }}
|data9 =                 {{#if:{{{location|}}}{{#ifeq:{{{nomap|}}}|yes|1|0}}|{{#display_points:
<{{{location|}}}>}} }}
|header20= {{#if:{{{cname|}}}| Contact}}
|data21  = {{Contact
 |name   = {{{cname|}}}
 |title  = {{{ctitle|}}}
 |child  = yes
 |image  = {{{cimage|}}}
 |email  = {{{cemail|}}}
 |site = {{{csite|}}}
 |phone  = {{{cphone|}}}
 |fax    = {{{cfax|}}}
 |address1= {{{caddress1|}}}
```

```
 |address2= {{{caddress2|}}}
 |address3= {{{caddress3|}}}
}} }} }}<noinclude>
{{Infobox Organization
 |name        = Organization Name
 |logo        = image.img
 |website  = example.com
 |founded  = 1000
 |location = space

 |cname         = Contact Name
 |cimage        = image.img
 |ctitle        = Title
 |csite =
 |cemail        = email@example.com
 |cphone        = (555) 555-5555
 |cfax          = (555) 555-5556
 |caddress1 = Address line 1
 |caddress2 = Address line 2
 |caddress3 = Address line 3

 <!--(optional fields:)-->
 |headerstyle  = ;
 |labelstyle   = ;
 |bodystyle    = ;
 |nomap = yes   <!--hides the map, even if there's an address/location-->
}}
<pre style="overflow:auto;">
{{Infobox Organization
 |name        =
 |logo        =
 |website  =
 |founded  =
 |location =
 |city =

 |cname         =
 |cimage        =
 |ctitle        =
 |cemail        =
 |csite         =
 |cphone        =
 |cfax          =
 |caddress1 =
 |caddress2 =
 |caddress3 =

 <!--(optional fields:)-->
 |headerstyle  = ;
 |labelstyle   = ;
 |bodystyle    = ;
 |nomap =      <!--hides the map when set to "yes", even if there's an address/location-->
}}
</pre>
</noinclude>
```

Template:Infobox LanguageProgram

```
{{#if:{{{name|}}}|{{Infobox
|bodystyle= width:30em;{{{bodystyle|}}};
|above = {{{name}}}
|image = {{#if:{{{logo|}}}|[[File:{{{logo}}}|300px|center]]}}
|headerstyle  = background:#cba;{{{headerstyle|}}};
|labelstyle   = background:#dcb;{{{labelstyle|}}};
|header1  = Program Information
|label2   = Website
|data2 =        {{#if:{{{website|}}}|       {{URL|{{{website}}}|{{{website}}}       }}       [[Has
website::{{{website}}}| ]]}}
|label3   = Founded
```

```
|data3 = {{#if:{{{founded|}}}|{{{founded|}}}[[Was founded::{{{founded|}}}| ]] }}
|label8  = {{#if:{{{location|}}}| Location}}
|data8 = {{Location|{{{location|}}} }}
|data9 = {{#if:{{{location|}}}|{{#display_points: <{{{location|}}}>}} }}
|header10 = {{#if:{{{language|}}}|Language Information}}
|label11  = {{#if:{{{language|}}}|Language}}
|data11               = {{#if:{{{language|}}}|{{#replace:{{{language}}}|&#59;|,     }}     [[Has
language::{{#replace:{{{language}}}|{{!}} {{)}}!}}{{!((}}Has language::}}| ]]
([http://sandbox.cityknowledge.net/wiki/index.php?search={{urlencode:{{{language}}}          }}
Directory Search])}}
|label12  = {{URL|http://www-01.sil.org/iso639-3/|ISO 639-3 code}}
|data12   = <code>{{{isocode|}}}</code>
|label13                = {{#if:{{{private|}}}|      Private?      [[Is      private
language::{{#ifeq:{{{private}}}|yes|no|Yes|No}}| ]]}}
|data13 = {{{private|}}}


|header20 = {{#if:{{{cname|}}}|Contact Information}}
|data21         = {{Contact
 |name   = {{{cname|}}}
 |title  = {{{ctitle|}}}
 |child  = yes
 |image  = {{{cimage|}}}
 |email  = {{{cemail|}}}
 |phone  = {{{cphone|}}}
 |fax    = {{{cfax|}}}
 |address1= {{{caddress1|}}}
 |address2= {{{caddress2|}}}
 |address3= {{{caddress3|}}}
 }} }} }} <noinclude>
{{Infobox LanguageProgram
 |name  = Program Name
 |logo  = image.img
 |website  = example.com
 |founded = 2010
 |location = space
 |language = all
 |private =
 |isocode  = all

 |cname        = Contact Name
 |cimage       = image.img
 |ctitle       = Title
 |cemail       = email@example.com
 |cphone       = (555) 555-5555
 |cfax         = (555) 555-5556
 |caddress1 = Address line 1
 |caddress2 = Address line 2
 |caddress3 = Address line 3

<!--(optional fields:)-->
 |headerstyle  = ;
 |labelstyle   = ;
 |bodystyle    = ;
}}
<pre style="overflow:auto;">
{{Infobox LanguageProgram
 |name =
 |logo =
 |website  =
 |founded =
 |location =
 |language =
 |private =
 |isocode  =

  |cname        =
  |cimage       =
  |ctitle       =
  |cemail       =
```

```
   |cphone    =
   |cfax     =
   |caddress1 =
   |caddress2 =
   |caddress3 =

<!--optional fields:-->
 |headerstyle  = ;
 |labelstyle   = ;
 |bodystyle    = ;
}}
</pre>
</noinclude>
```

Template:Fragment

```
{{Ambox
|type = content
|text = This Directory entry contains '''very limited information'''. If this is your page, it is
recommended that you [[NLRDirectory:Entry wizard|contact the site administrators]] about updating
it.
|small = no
}}
[[Category:Entries with very limited information]]
```

Template:Incomplete

```
{{Ambox
|type = content
|text = This Directory entry '''is missing information''', making it incomplete. If this is your
page, it is recommended that you [[NLRDirectory:Entry wizard|contact the site administrators]] to
update it.
|small = no
}}
[[Category:Entries with incomplete content]]
```

Template:Langprivacy

```
{{#ifeq:{{#show:{{PAGENAME}}|?Is   private   language}}|true|{{Ambox|text=This   community   does   not
teach its language to outsiders.|type=notice}} }}
```

Template:ELP

```
[http://www.endangeredlanguages.com{{#if:{{#show:{{PAGENAME}}|?Has
language=}}|/lang/search/#/?q={{urlencode:{{#show:{{PAGENAME}}|?Has   language=}} }}|  }}    The
Endangered    Langauges    Project{{#if:{{#show:{{PAGENAME}}|?Has    language=}}|'s   Information   on
{{#show:{{PAGENAME}}|?Has language= }} }}]<noinclude>

If the page this is included on has a <nowiki>[[Has language::]]</nowiki> tag embedded in it, the
link will point to the ELP's search page for that language, and state it in the link text.

If not, the link merely goes to the ELP's home page.
</noinclude>
```

Template:Location

```
{{#if:{{{1|}}}|{{{1}}}[[Has      street      address::{{{1}}}|    ]][[Has      coordinates::{{#geocode:
{{{1}}}|service=google}}| ]] }}<noinclude>

This is a shortcut to make two Semantic MediaWIki tags based on an address.

==Example Usage==
<pre>{{Location|New York, NY}}</pre>
==Result==
'''Displays:'''
::New York, NY
```

```
'''Tags:'''
:: ''Has address:'' New York, NY
:: ''Has coordinates:'' {{#geocode: New York, NY|service=google}}


</noinclude>
```

Customized Templates

These templates have been heavily modified to the point that they warrant mention. They have been adapted from the Wikipedia templates of the same names and either specialized or fixed to cooperate with the Directory's MediaWiki install.

Template:Article wizard

```
<div style="text-align:center;border-bottom:1px solid #9accf6">
{{Article wizard/box|num=1|link=|label=Introduction|do={{{1|}}} }}
{{Article wizard/box|num=2|link=Guidelines|label=Content Guidelines|width=20%|do={{{2|}}} }}
{{Article wizard/box|num=3|link=Your entry|label=Your Entry|do={{{3|}}} }}
<div style="width:0;height:0;clear:both;overflow:hidden"></div>
</div>
<div style="border:1px solid #9accf6;padding:.5em 1em 1em 1em;border-top:none">
{{{content}}}
<div style="width:0;height:0;clear:both;overflow:hidden"></div> </div>


[[Category:Directory              entry              wizard]]{{#if:{{PROTECTIONLEVEL:edit}}|{{pp-
protected|small=yes}}}}<noinclude>
== Usage ==
<pre style ="overflow:auto;">
{{Article wizard|1=type|2=type|3=type|content=content}}
</pre>
where
*type is one of '''plain''', '''linked''', '''italic''', '''boxed'''
*content is the content of the page which appears inside the box


==Example==
<nowiki>{{Article wizard|1=linked|2=italic|3=boxed|content=Hello. This is a test.}}</nowiki>
produces
{{Article wizard|1=linked|2=italic|3=boxed|content=Hello. This is a test.}}

</noinclude>
```

Template:Article wizard/box

```
{{#switch:{{{do}}}   |italic=<div style="float:left;width:{{#if:{{{width|}}}|{{{width}}}|12.6%}};
font-size:1.4em; {{border-radius | 4px 4px 0px 0px}} margin:1px 1px 0; line-height: 2em;margin-
bottom: -0.2em;  background-color:#F4F4F4;color:#000;border:1px solid  #B4B4B4;border-bottom:0;
color:#6D6D6D;">                       ''{{{label}}}''</div>                    |linked=<div
style="float:left;width:{{#if:{{{width|}}}|{{{width}}}|12.6%}}; font-size:1.4em; {{border-radius
| 4px    4px    0px    0px}}    margin:1px    1px    0;    line-height:   2em;background-
color:#f1f9ff;color:#000;border:1px    solid    #9accf6;border-bottom:0">    [[NLRDirectory:Entry
wizard{{#if:{{{link|}}}|/}}{{{link}}}|{{{label}}}]]</div>                    |boxed=<div
style="float:left;width:{{#if:{{{width|}}}|{{{width}}}|12.6%}}; font-size:1.4em; {{border-radius
| 4px  4px  0px  0px}} margin:1px  1px  0;  line-height:  2em;margin-bottom:-0.3em;font-
weight:bold;background-color:#fff;color:#000;border:1px    solid    #9accf6;border-bottom:0;   line-
height:           2.1em;">            {{{label}}}</div>                         |plain=<div
style="float:left;width:{{#if:{{{width|}}}|{{{width}}}|12.6%}}; font-size:1.4em; {{border-radius
| 4px 4px 0px 0px}} margin:1px 1px 0; line-height: 2em;">{{{label}}}</div> |#default=<div
style="float:left;width:{{#if:{{{width|}}}|{{{width}}}|12.6%}}; font-size:1.4em; {{border-radius
| 4px 4px 0px 0px}} margin:1px 1px 0; line-height: 2em;margin-bottom: -0.2em;background-
color:#F4F4F4;color:#000;border:1px   solid   #B4B4B4;border-bottom:0;   color:#6D6D6D;  opacity:
0.30;"> {{{label}}}</div>}}
```

Appendix C: Directory Entry Generation Script

File: generator.py

```
#!C:\python26\python.exe


# CSV to XML converter
# created for the 2013 Santa Fe IQP group working with the Indigenous Language Institute



from os         import fsync
from os.path    import abspath, exists, getsize, isdir
from time       import gmtime, strftime
from sys        import argv,exit,stdout
import csv
import formatter


verbose = 1
debug   = 1


sscol_entry_type = 1
sscol_entry_status = 2
sscol_entry_more = 3
sscol_organization = 4
languageprog = "LP"
organization = "OR"
funding      = "FS"


MAX_PAGE_COUNT = 40


#   Column definitions
cols = {
" "                 : 0,
"organization"      : 4,
"language"          : 5,
"isprivate"         : 6,
"isocode"           : 7,
"name"              : 8,
"overview"          : 9,
"description"       : 10,
"tag_immersion"     : 11,
"tag_edufund"       : 12,
"tag_adultclass"    : 13,
"tag_faminvolve"    : 14,
"tag_materials"     : 15,
"tag_prog_assist"   : 16,



"contact_phone"     : 17,
"contact_fax"       : 18,
"contact_address"   : 19,
"contact_city"      : 20,
"contact_state"     : 21,
"contact_country"   : 22,
"contact_continent" : 23,
"website"           : 24,
"website_perprogram" : 25,
"contact_email"     : 26,
"contact_name"      : 27,
"contact_position"  : 28}



#main categories
cat_fund = "Funding Sources"
```

```
cat_prog = 'Language Programs'
cat_org  = "Organizations"


defaultBlurb = "This is the Directory entry for {0}."
noinfo = "x"
good = "g"
part = "i"
bad  = "x"
unk  = "u"
dupe = "d"
nocurrentlangs = "ncr"


def smartGetEntry(row, col, test=0, LP=0):
        data = ""
        if type(col) == type(" "):
        data = row[cols[col]]
        elif type(col)== type(1):
        data = row[col]

        if data is None:
        return ""

        data = data.strip()#.replace("\n", ",")

        if data.lower() == noinfo:
        data = ""

        #lines that should get newlines removed. Like titles
        if col in [sscol_organization, "contact_address"]:
        tmp = data.split("\n")
        for i in range(len(tmp)):
                tmp[i] = tmp[i].strip()

        data = ", ".join(filter(None, tmp))

        if data != "" and col == sscol_entry_type:
        if data == languageprog:
                data = cat_prog
        if data == organization:
                data = cat_org
        if data == funding:
                data = cat_fund
        if data == "" and col == "contact_name":
        data = row[cols["organization"]]
        if data == "" and col == "overview":
        if LP and (not smartGetEntry(row, "name", 1)) and smartGetEntry(row, "description", 1):
                data = smartGetEntry(row, "description")
        else:
                data = defaultBlurb.format(smartGetEntry(row, sscol_organization))

        if test:
        data = data.lower()
        return data


def nextfile(outpath, count, prevfile=None):
        if prevfile:
        prevfile.write(formatter.base.split("  {page}\n")[-1])
        prevfile.flush()
        fsync(prevfile)
        prevfile.close()
        verprint("Closing file "+outpath.format(str(count-1)))

        if count>0:
        verprint("Opening file "+outpath.format(str(count))+" for output")
        output = open(outpath.format(str(count)), "w+")
        output.write(formatter.base.split("  {page}\n")[0])
        return output
```

```
def go(csvfile, nosave = 0, noGeocode = 0):
        filecount = 1
        outpath = abspath("./output")+"/XMLgen-"+\
                        strftime("%Y-%m-%dT%H-%M-%SZ", gmtime())+\
                        "-{0}.xml"
        output = None


        verprint("Opening CSV "+abspath(csvfile))
        linelist = csv.reader(open(csvfile, "rb"))
        #verprint("CSV contains "+len(linelist)+" lines")

        if not nosave:
        output = nextfile(outpath, filecount)


        contents = formatter.base.format(\
        site    = "localhost",\
        mainpage = "localhost/wiki/Main_Page",\
        generator= "Custom, Python 2.6.6, Win7x86/",\
        page    = formatter.nextpage)

        prevline = linelist.next()
        verprint("Found the following columns:\n"+str(prevline))
        line = linelist.next()

        #program matches next program?
        #Y: add data from line, go around again
        #N: finish off entry, start another with the next one

        #if not nosave:

        #initialize lots of variables

        progcount = 0
        orgcount  = 0
        fundcount = 0

        goodcount = 0
        ehcount   = 0
        badcount  = 0

        entrycount = 0
        pagecount  = 0
        blankcount = 0

        citylist = []
        statelist = []
        countrylist = []
        continentlist = []
        city_geolocations = []
        smwpagecount = 0
        SMW_tagpage_count  = 0
        extrapages_made = []
        SMW_citypage_count = 0
        SMW_statepage_count = 0
        SMW_countrypage_count = 0
        SMW_continentpage_count = 0

        SMW_tag_count = 0

        geocodeErr = ""

        #potentially cross-iteration states
        currentLineType = ""
```

```
        programs = 0
        article  = ""
        hasinfo  = 0


        # == Main page generation routine ====================================================
        while prevline:
        #lines+=1
        entrycount+=1


        if pagecount and pagecount%MAX_PAGE_COUNT is 0:
                if not nosave:
                filecount+=1
                output = nextfile(outpath, filecount, output)
                else:
                verprint("would be moving on to file number "+str(filecount))


        #verprint("Processing entry # "+str(entrycount))#+"  ("+smartGetEntry(line,
sscol_organization].strip()+")")


        #blank line (eg: ,,,,,,,,,)
        if (not line) or smartGetEntry(line, sscol_organization) == '': #or line.count('')>=10:
                prevline = line
                try:
                line = linelist.next()
                except:
                line = None
                verprint("Entry #"+str(entrycount)+"  doesn't have enough data. Skipping...")
                blankcount+=1
                continue


        #no line is next (EOF)
        #no different line is next (names don't match)
        # so we end this page and start the next one
        elif ((not line) or (smartGetEntry(line, sscol_organization, 1) != smartGetEntry(prevline,
sscol_organization, 1))):


                # We'll create a page as long as we have a name for it. Otherwise it's skipped.
                if smartGetEntry(prevline, sscol_organization):
                verprint("Found all entries for '"+smartGetEntry(prevline, sscol_organization)+"'.
Generating XML.")


                article+=formatter.othersites+\
                        (formatter.references if article.count("</ref>")>0 else "")+\
                        formatter.categorybox.format(currentLineType)


                if not nosave:
                        output.write(formatter.populateArticleXML(smartGetEntry(prevline,
sscol_organization), article, str(pagecount)))


                pagecount+=1
                else:
                verprint("Found no program name. Skipping.")


                hasinfo = 0
                programs = 0


                #print article
                #article+=formatter.h2.format(smartGetEntry(line, "name")+\
                #        smartGetEntry(line, "description"))


        else:
                verprint("Appending information to current entry")


        #next program is the same as this one, so we just add to the article.
        #else:#if smartGetEntry(line, sscol_organization) is nextline[sscol_organization]:
```

```python
if not programs and smartGetEntry(line, sscol_entry_more, 1) != dupe:
        verprint("Starting entry for "+smartGetEntry(line, sscol_organization))
        article  = ""

        verprint("Article status: "+smartGetEntry(line, sscol_entry_status)+\
               "/"+smartGetEntry(line, sscol_entry_more))
        if smartGetEntry(line, sscol_entry_status, 1) == part:
        article += formatter.limited_info
        verprint("Entry #"+str(entrycount)+" has incomplete info. Marking...")
        ehcount +=1
        elif smartGetEntry(line, sscol_entry_status, 1) in [ bad, unk, ""]:
        article += formatter.limited_info
        verprint("Entry #"+str(entrycount)+" has a minimum of info. Marking...")
        badcount+=1
        else:
        goodcount+=1

        if smartGetEntry(line, sscol_entry_more) == nocurrentlangs:
        article+=formatter.no_current_langs

        currentLineType = smartGetEntry(line, sscol_entry_type)

        #some of the "contact emails" are actually links to contact forms on websites.
        #this handles those
        email = smartGetEntry(line, "contact_email")
        site  = ""
        if not (email and email.count("@") and not email.count("http://")):
        site = email
        email = ""

        #this handles the various different types of entries we have.
        if currentLineType == cat_prog: # Language Programs ===========================
        progcount+=1
        article+=formatter.programentry
        article = article.format(website  = smartGetEntry(line, "website"),\
                    location = smartGetEntry(line, "contact_address"),\
                    lang   = smartGetEntry(line, "language"),\
                    isocode  = smartGetEntry(line, "isocode"),\
                    person   = smartGetEntry(line, "contact_name"),\
                    title  = smartGetEntry(line, "contact_position"),\
                    email  = email,\
                    csite  = site,\
                    phone  = smartGetEntry(line, "contact_phone"),\
                    fax      = smartGetEntry(line, "contact_fax"),\
                    address1 = "",\
                    address2 = "",\
                    address3 = "",\
                    founded  = "",\
                    blurb  = smartGetEntry(line, "overview", LP=1))

        elif currentLineType == cat_fund:# Funding Source ===========================
        fundcount+=1
        article+=formatter.fundingentry
        article = article.format(website  = smartGetEntry(line, "website"),\
                    location = smartGetEntry(line, "contact_address"),\
                    person   = smartGetEntry(line, "contact_name"),\
                    title  = smartGetEntry(line, "contact_position"),\
                    email  = email,\
                    csite  = site,\
                    phone  = smartGetEntry(line, "contact_phone"),\
                    fax      = smartGetEntry(line, "contact_fax"),\
                    address1 = "",\
                    address2 = "",\
                    address3 = "",\
                    founded  = "",\
                    blurb  = smartGetEntry(line, "overview"))
```

```
        elif currentLineType == cat_org: # Organization ==============================
        orgcount+=1
        article+=formatter.organizationentry
        article = article.format(website  = smartGetEntry(line, "website"),\
                        location = smartGetEntry(line, "contact_address"),\
                        person   = smartGetEntry(line, "contact_name"),\
                        title    = smartGetEntry(line, "contact_position"),\
                        email    = email,\
                        csite    = site,\
                        phone    = smartGetEntry(line, "contact_phone"),\
                        fax      = smartGetEntry(line, "contact_fax"),\
                        address1 = "",\
                        address2 = "",\
                        address3 = "",\
                        founded  = "",\
                        blurb    = smartGetEntry(line, "overview"))


        citylist.append([smartGetEntry(line, "contact_city"),
                        smartGetEntry(line, "contact_state"),
                        smartGetEntry(line, "contact_country"),
                        smartGetEntry(line, "contact_continent")])
        #print "\t\t Appending "+str(citylist[-1])+" to list of cities to generate"

        programs+=1

        if currentLineType == cat_prog:
        #LP with a named program
        if smartGetEntry(line, "name"):
                hasinfo = 1
                article+=formatter.programs_offered
        #LP with an unnamed program --now handled by smartGetEntry in the blurb
        #elif smartGetEntry(line, "description"):
        #       article+=smartGetEntry(line, "description")
        #LP with no known data
        else:
                verprint("Entry "+str(entrycount)+"has no program information")

    if hasinfo:
        article+="\n"+formatter.h2.format(smartGetEntry(line, "name"))+\
                smartGetEntry(line, "description")


    prevline = line
    try:
        line = linelist.next()
    except:
        line = None


# == Semantic MediaWiki page generation section =========================================
# ===== Geographics. Cities, states, countries. =========================================
for city in citylist:
    if pagecount and pagecount%MAX_PAGE_COUNT is 0:
        if not nosave:
        filecount+=1
        output = nextfile(outpath, filecount, output)
        else:
        verprint("would be moving on to file number "+str(filecount))

    # a city in the list of cities has not yet had a page created for it
    if city[0] and not extrapages_made.count(city[0].lower()):
        pagecount += 1
        SMW_citypage_count+=1
        #verprint("creating page for "+city[0]+", "+formatter.expandstate(city[1]))
        if not nosave:
        output.write(formatter.make_placepage(city[0], pagecount, "city", city[1]))
```

```
                    extrapages_made.append(city[0].lower())  #Append  the  entry  to  the  list  of  pages
created, to avoid dupes

                    #Only try to geocode if 0. All nonzero values eval to False.
                    #this allows for errors and disables to be handled
                    if not noGeocode:
                    try:
                            citystate  =  city[0]+",  "+(city[1]  if  city[2]  ==  "United  States"  else
city[2])
                            verprint("\tMaking Google Maps API request to geocode '"+citystate+"'")
                            city_geolocations.append([citystate, formatter.geocode(citystate)])
                            verprint("\tGot coodinates: "+str(city_geolocations[-1]))
                    except Exception as e:
                            geocodeErr = "Error ({0}): {1}".format(type(e), e.message)
                            noGeocode = -1

        #A state or province associated with a city has not yet had a page created for it
        if city[1] and not extrapages_made.count(city[1].lower()):
                pagecount += 1
                SMW_statepage_count+=1
                if city[2] == "United States":
                verprint("creating page for "+city[2]+" state "+formatter.expandstate(city[1]))
                if not nosave:
                        output.write(formatter.make_placepage(formatter.expandstate(city[1]),
pagecount, "state", city[2]))
                else:
                verprint("creating page for "+city[2]+" province "+city[1])
                if not nosave:
                        output.write(formatter.make_placepage(formatter.expandstate(city[1]),
pagecount, "province", city[2]))
                extrapages_made.append(city[1].lower())  #Append  the  entry  to  the  list  of  pages
created, to avoid dupes


        # A country associated with the list of
        if city[2] and not extrapages_made.count(city[2].lower()):
                pagecount += 1
                SMW_countrypage_count+=1
                verprint("creating  page  for  "+city[2]+("n"  if  city[3].endswith("a")  else  "an")+"
country "+city[2])
                if not nosave:
                output.write(formatter.make_placepage(city[2], pagecount, "country", city[3]))
                extrapages_made.append(city[2].lower())


        SMW_page_count= SMW_tagpage_count+\
                        SMW_citypage_count+\
                        SMW_statepage_count+\
                        SMW_countrypage_count+\
                        SMW_continentpage_count


        # == Close the last open file and create staaaaaats ====================================
        if not nosave:
        nextfile(outpath, 0, output)
        print "\n\n"
        verprint("Done.\n")
        print "\n"+(" Stats ").center(78, "=")
        print "Source: "+csvfile
        print "          ("+str(getsize(csvfile)/1024.00)+" kB)"
        print "Output: "
        for n in range(filecount):
        print " "*8+outpath.format(n+1)
        if not nosave:
                print "          ("+str(getsize(outpath.format(n+1))/1024.00)+" kB)"
        else:
                print "          (nosave = 1; document unsaved)"


        print (str(entrycount-blankcount).rjust(9)+" entries in")
        print "|".rjust(11)+str(blankcount).rjust(5)+" blank entries"
        print "|".rjust(11)+str(goodcount).rjust(5)+" mostly complete entries"
        print "|".rjust(11)+str(ehcount).rjust(5)  +" partially complete entries"
```

```
        print "|".rjust(11)+str(badcount).rjust(5) +" entries missing lots of data"

        print (str(pagecount-SMW_page_count).rjust(9)+" content pages out")
        print "|".rjust(11)+str(progcount).rjust(5)+" program pages"
        print "|".rjust(11)+str(orgcount).rjust(5) +" organization pages"
        print "|".rjust(11)+str(fundcount).rjust(5)+" funding source pages"

        print str(SMW_page_count).rjust(9)+" Auxiliary Semantic MediaWiki-related pages:"
        print  "|".rjust(11)+str(SMW_tagpage_count).rjust(5)+" Semantic  MediaWiki  tag  detail
pages"
        print "|".rjust(11)+str(SMW_citypage_count).rjust(5)+" SMW city detail pages"
        print "|".rjust(11)+str(SMW_statepage_count).rjust(5)+" SMW state detail pages"
        print "|".rjust(11)+str(SMW_countrypage_count).rjust(5)+" SMW country detail pages"
        print "|".rjust(11)+str(SMW_continentpage_count).rjust(5)+" SMW continent detail pages"


        if noGeocode<0:
        print " !!"+(" "*4)+"An error occurred while trying to geocode data:\n\t\t"+geocodeErr
        if noGeocode is 0:
        print ("   "*7)+"Successfully  geocoded   "+str(len(city_geolocations))+"    locations"
        #Geocoding stats
        if noGeocode>0:
        print (" "*7)+"Geocoding was disabled for this run."

        print "\n"+str(SMW_tag_count).rjust(9)+" SMW tags added to pages"

        print "\n"+str(pagecount).rjust(9)+" Total pages created."

        #print "detailed city list:"+repr(citylist)


def verprint(msg, level=1):
        if verbose>=level:
        print strftime("%H:%M:%S::", gmtime())+"  "+msg

def exitmsg(msg):
        print msg

#argument parser for startup
if __name__ == "__main__":

        args = argv[1:]
        if not args:
        exit(exitmsg("We need something here"))

        if isdir(args[0]):
        exit(exitmsg("Error: "+args[0]+" is a directory"))

        if not exists(args[0]):
        exit(exitmsg("Error: "+args[0]+" is not available"))

        filepath = abspath(args[0])

        #start the show
        go(args[0])
        exit(0)
```

File: formatter.py

```
import hashlib
from base64          import b32encode as encode
from httplib         import HTTPConnection
from json            import loads as json_loads
```

```
from time           import gmtime, strftime
from urllib         import urlencode
from xml.sax.saxutils import escape, unescape

statelist = {           #US and territories
        'AK': 'Alaska',         'AL': 'Alabama',        'AR': 'Arkansas',
        'AS': 'American Samoa', 'AZ': 'Arizona',        'CA': 'California',
        'CO': 'Colorado',       'CT': 'Connecticut',    'DC': 'District of Columbia',
        'DE': 'Delaware',       'FL': 'Florida',        'GA': 'Georgia',
        'GU': 'Guam',           'HI': 'Hawaii',         'IA': 'Iowa',
        'ID': 'Idaho',          'IL': 'Illinois',       'IN': 'Indiana',
        'KS': 'Kansas',         'KY': 'Kentucky',       'LA': 'Louisiana',
        'MA': 'Massachusetts',  'MD': 'Maryland',       'ME': 'Maine',
        'MI': 'Michigan',       'MN': 'Minnesota',      'MO': 'Missouri',
        'MS': 'Mississippi',    'MT': 'Montana',        'NC': 'North Carolina',
        'ND': 'North Dakota',   'NE': 'Nebraska',       'NH': 'New Hampshire',
        'NJ': 'New Jersey',     'NM': 'New Mexico',     'NV': 'Nevada',
        'NY': 'New York',       'OH': 'Ohio',           'OK': 'Oklahoma',
        'OR': 'Oregon',         'PA': 'Pennsylvania',   'PR': 'Puerto Rico',
        'RI': 'Rhode Island',   'SC': 'South Carolina', 'SD': 'South Dakota',
        'TN': 'Tennessee',      'TX': 'Texas',          'UT': 'Utah',
        'VA': 'Virginia',       'VI': 'Virgin Islands', 'VT': 'Vermont',
        'WA': 'Washington',     'WI': 'Wisconsin',      'WV': 'West Virginia',
        'WY': 'Wyoming',        #and now Canada
        'AB': 'Alberta',        'BC': 'British Columbia',
        'MB': 'Manitoba',       'NB': 'New Brunswick',  'NL': 'Newfoundland and Labrador',
        'NT': 'Northwest Territories',                  'NS': 'Nova Scotia',
        'NU': 'Nunavut',        'ON': 'Ontario',        'PE': 'Prince Edward Island',
        'QC': 'Quebec',         'SK': 'Saskatchewan',   'YT': 'Yukon'
}

def geocode(location=None):
        if location:
        location = location.replace("\n", " ")
        query=urlencode({'q':location})


        gmaps=HTTPConnection("maps.googleapis.com")
        gmaps.request("GET","/maps/api/geocode/xml?address="+query+"&sensor=false")
        print                                   "\t\t                              Request:
maps.googleapis.com/maps/api/geocode/json?address="+query+"&sensor=false"
        search=gmaps.getresponse()
        data=search.read()

        latlong = json_loads(data)["results"][0]["geometry"]["location"]
        #status  = json_loads(data)["status"]

        #print "\t\t Status: "+status

        return [latlong["lat"], latlong["lng"]]

def getElement(entry, tag):
        try:
        return entry.getElementsByTagName(tag)[0].childNodes[0].nodeValue
        except:
        return u''

#Calculate the length of the article in bytes
#   This would just be the number of characters, but expanded HTML
#      (eg '&amp;' for '&') counts differently.
def articleLength(article):
        return len(unescape(article))

#Generate a SHA-1 message digest based on the supplied article
def digest(article):
        sha = hashlib.sha1(article)
        digest = sha.digest()
```

```python
        del sha
        return encode(digest)


def HTMLize(article):
        return escape(article)


def populateArticleXML(title, article, aID, rID="1", prID="0", lastEditor="admin", uID="1",
comment = "automated page creation"):
        namespace='0'
        if title.startswith("Template:"):
        namespace = '10'
        elif title.startswith("Category:"):
        namespace = '14'
        elif title.startswith("Property:"):
        namespace = '102'

        return page.format(title       = title                  \
                ,namespace      = namespace              \
                ,articleid      = aID                    \
                ,revid          = rID                    \
                ,rev_parentid   = prID                   \
                ,rev_timestamp  = strftime("%Y-%m-%dT%H:%M:%SZ", gmtime()) \
                ,rev_uname      = lastEditor             \
                ,userid         = uID                    \
                ,comment        = comment                \
                ,nbytes         = len(article)           \
                ,article        = HTMLize(article))#     \
#               ,sha            = digest(article)


def appendArticleXML(xml, article, final=0):

        if final:
        xml = xml.format(page=article)
        else:
        xml = xml.format(page=article+nextpage)

        return xml


def smwtag(propname, propval, isattrib=0, hide=0):
        return begintag+propname+\
                (tag_delim_attrib if isattrib else tag_delim_property)+\
                propval+(hidetag if hide else "")+endtag


def make_proppage(propname, id, type):
        return   populateArticleXML("Property:"+propname,   "This   is   a   property   of   type
"+proptypes[type], id)


def make_placepage(placename, id, type, parent):
        return populateArticleXML(placename, placepages[type].format(parent), id)


def expandstate(stateid):
        try:
        return statelist[stateid]
        except:
        print "\t\t Could not successfully expand "+stateid
        return stateid


#Semantic MediaWiki tags  ==========================================================================


placepages = {
"city"      : "{{{{PAGENAME}}}} is a city in [[Located in::{0}]].",
"state"     : "{{{{PAGENAME}}}} is a state in [[Located in::{0}]].",
"province"  : "{{{{PAGENAME}}}} is a province of [[Located in::{0}]].",
"country"   : "{{{{PAGENAME}}}} is a country in [[Located in::{0}]]."}


proptypes = {
```

```
"bool" :"[[Has type::Boolean]]",
"code" :"[[Has type::Code]]",
"date" :"[[Has type::Date]]",
"email":"[[Has type::Email]]",
"geo"  :"[[Has type::Geographic coordinate]]",
"num"  :"[[Has type::Number]]",
"page" :"[[Has type::Page]]",
"quan" :"[[Has type::Quantity]]",
"text" :"[[Has type::Text]]",
"phone":"[[Has type::Telephone number]]",
"url"  :"[[Has type::URL]]"}

props = {
"has"   : "Has type",
"in"    : "Located in",
"coords": "Has coordinates",
"loc"   : "Has location type",
"subprop":"Subproperty of",
"site"  : "website"
}

begintag = "[["
tag_delim_attrib   = ":="
tag_delim_property = "::"
hidetag  = "| "
endtag   = "]]"




#page elements          ============================================================
h1 = "=={0}==\n"
h2 = "==={0}===\n"
h3 = "===={0}====\n"


bullet = "* "


programs_offered = "==Programs  Offered==\n{{langprivacy}}\n{{PAGENAME}}  offers  the  following
programs:"
things_funded = "==Funded  and  Supported  Programs==\n{{PAGENAME}}  funds  and/or  supports  the
following programs:"


seealso = "\n==See Also==\n"
othersites = """\n==External Links==
* [http://www.endangeredlanguages.com Endangered Languages Project]
*    [http://www.unesco.org/culture/en/endangeredlanguages/    UNESCO:    Endangered    Languages
Programme]"""


"""   *   [[http://www.endangeredlanguages.com/lang/search/#/?q={search}   Endangered   Languages
Project]]"""


no_current_langs = "{{{{ambox |text=This Organization is currently not known to be  associated
with any language programs.}}}}"
language_privacy = "{{{{langprivacy}}}}"
limited_info   = "{{{{incomplete}}}}"
minimal_info   = "{{{{fragment}}}}"


references = "\n==References==\n<references/>"


categorybox = "\n[[Category:{0}]]"


#big chunks of pages          =================================================================


# Tags:
#  {title}           Page title (Templates still start with "Template:"
#  {namespace}       Namespace of the page. Regular pages are 0, templates 10, categories 14.
```

```
#   {articleid}        ID of article
#   {revid}            ID of current revision
#   {rev_parentid}     ID of previous revision
#   {rev_timestamp}    Timestamp of last revision; format is: 2013-03-15T16:45:44Z
#   {rev_uname}        Username of last revisor
#   {userid}           User ID of last revisor
#   {comment}          Revision comment
#   {nbytes}           Size of article in bytes (number of characters, aside from the size of
HTML chars (eg " ")
#   {article}          Article contents
#   {sha}              SHA1 checksum


page = """\
  <page>
        <title>{title}</title>
        <ns>{namespace}</ns>
        <id>{articleid}</id>
        <revision>
        <id>{revid}</id>
        <parentid>{rev_parentid}</parentid>
        <timestamp>{rev_timestamp}</timestamp>
        <contributor>
        <username>{rev_uname}</username>
        <id>{userid}</id>
        </contributor>
        <minor/>
        <comment>{comment}</comment>
        <text xml:space="preserve" bytes="{nbytes}">{article}</text>
        <model>wikitext</model>
        <format>text/x-wiki</format>
        </revision>
  </page>
"""
#after <text/>:         <sha1>{sha}</sha1>


nextpage = """{page}"""


#Based on the model pages:

fundingentry = """{{{{Infobox Organization
 |name  = {{{{PAGENAME}}}}
 |logo =
 |website  = {website}
 |founded  = {founded}
 |location = {location}
 |cname        = {person}
 |cimage       =
 |ctitle       = {title}
 |cemail       = {email}
 |csite        = {csite}
 |cphone       = {phone}
 |cfax         = {fax}
 |caddress1 = {address1}
 |caddress2 = {address2}
 |caddress3 = {address3}
}}}}


{blurb}


"""


programentry = """{{{{Infobox LanguageProgram
 |name  = {{{{PAGENAME}}}}
 |logo =
 |website  = {website}
 |location = {location}
 |language = {lang}
 |isocode  = {isocode}
```

```
   |private  =

 |cname       = {person}
 |cimage      =
 |ctitle      = {title}
 |cemail      = {email}
 |csite       = {csite}
 |cphone      = {phone}
 |cfax        = {fax}
 |caddress1 = {address1}
 |caddress2 = {address2}
 |caddress3 = {address3}
}}}}


{blurb}


"""


organizationentry = """{{{{Infobox Organization
 |name = {{{{PAGENAME}}}}
 |logo =
 |website  = {website}
 |founded  = {founded}
 |location = {location}

 |cname       = {person}
 |cimage      =
 |ctitle      = {title}
 |cemail      = {email}
 |csite       = {csite}
 |cphone      = {phone}
 |cfax        = {fax}
 |caddress1 = {address1}
 |caddress2 = {address2}
 |caddress3 = {address3}
}}}}


{blurb}


"""


#  {site}          Name of Source MediaWiki
#  {mainpage}      Main        page        of        source       MediaWiki       -       eg
"http://en.wikipedia.org/wiki/Main_Page"
#  {generator}     Program that generated the page
#  {page}          Content of one page
base = """\
<mediawiki                                    xmlns="http://www.mediawiki.org/xml/export-0.8/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.mediawiki.org/xml/export-0.8/ http://www.mediawiki.org/xml/export-
0.8.xsd" version="0.8" xml:lang="en">
  <siteinfo>
      <sitename>{site}</sitename>
      <base>{mainpage}</base>
      <generator>{generator}</generator>
      <case>first-letter</case>
      <namespaces>
      <namespace key="-2" case="first-letter">Media</namespace>
      <namespace key="-1" case="first-letter">Special</namespace>
      <namespace key="0" case="first-letter"/>
      <namespace key="1" case="first-letter">Talk</namespace>
      <namespace key="2" case="first-letter">User</namespace>
      <namespace key="3" case="first-letter">User talk</namespace>
      <namespace key="4" case="first-letter">LanguageRevitapedia</namespace>
      <namespace key="5" case="first-letter">LanguageRevitapedia talk</namespace>
      <namespace key="6" case="first-letter">File</namespace>
      <namespace key="7" case="first-letter">File talk</namespace>
      <namespace key="8" case="first-letter">MediaWiki</namespace>
```

53

```
            <namespace key="9" case="first-letter">MediaWiki talk</namespace>
            <namespace key="10" case="first-letter">Template</namespace>
            <namespace key="11" case="first-letter">Template talk</namespace>
            <namespace key="12" case="first-letter">Help</namespace>
            <namespace key="13" case="first-letter">Help talk</namespace>
            <namespace key="14" case="first-letter">Category</namespace>
            <namespace key="15" case="first-letter">Category talk</namespace>
            <namespace key="102" case="first-letter">Property</namespace>
            <namespace key="103" case="first-letter">Property talk</namespace>
            <namespace key="108" case="first-letter">Concept</namespace>
            <namespace key="109" case="first-letter">Concept talk</namespace>
            <namespace key="420" case="first-letter">Layer</namespace>
            <namespace key="421" case="first-letter">Layer talk</namespace>
        </namespaces>
    </siteinfo>
    {page}
</mediawiki>
"""

organizationbox = """{{{{Infobox Organization
  |name        = {name}
  |logo        = {logo}
  |website  = {website}
  |founded  = {founded}
  |location = {location}


  |cname        = {person}
  |cimage       = {image}
  |ctitle       = {title}
  |cemail       = {email}
  |cphone       = {phone}
  |cfax         = {fax}
  |caddress1 = {address1}
  |caddress2 = {address2}
  |caddress3 = {address3}


  <!-- optional fields: -->;
  |headerstyle  = ;
  |labelstyle   = ;
  |bodystyle    = ;
  |nomap =
 }}}}
"""


languagebox = """{{{{Infobox LanguageProgram
 |name  = {name}
 |logo = {logo}
 |website  = {website}
 |location = {location}
 |language = {language}
 |isocode  = {iso}

 <!-- optional fields: -->
 |headerstyle  = ;
 |labelstyle   = ;
 |bodystyle    = ;
}}}}
"""


infobox = """\
 {{{{Infobox {type}
 |bodystyle    = width:25em;
 |above        = {label}

 |headerstyle  = background:#cba;
 |labelstyle   = background:#dcb;

 {nextline}
```

```
 }}}}"""


infobox_fields = {\
"label" : "|label{n} = {content}\n",\
"data"  : "|data{n}  = {content}\n",\
"header": "|header{n} = {content}\n"}


infobox_nextline = "\n{nextline}"


contact = """{{{{Contact\
 |who   = {name}
 |title = {title}


 |image  = {image}


 |email = {email}
 |phone = {phone}
 |fax   = {fax}


 |address1 = {address1}
 |address2 = {address2}
 |address2 = {address3}
 }}}}"""
```

Program Output

```
06:50:47::    Opening  file  /Volumes/Home/Documents/Dropbox/ID2050/xmlgen/output/XMLgen-2013-04-
30T06-50-47Z-1.xml for output
06:50:47::  Found the following columns:
['Timestamp', 'Type', 'Status', 'More', 'Untitled Question', 'Language', 'Private?', 'ISO 639-3
Code', 'Program Name', 'Program Overview (Goals)', 'Program Description', 'Immersion?',
'Educational Funds?', 'Adult Classes?', 'Family Involvement?', 'Materials?', 'Assists other
programs', 'Phone Number', 'Fax Number', 'Mailing Address', 'City', 'State', 'Country',
'Continent', 'Website Address', 'Per-Program website', 'Email Address', 'Contact Name', 'Contact
Position']
06:50:47::  Entry #1  doesn't have enough data. Skipping...
06:50:47::  Found no program name. Skipping.
06:50:47::  Starting entry for Absentee Shawnee Language Class
06:50:47::  Article status: G/
06:50:47::  Appending information to current entry
06:50:47::  Appending information to current entry
06:50:47::  Appending information to current entry
06:50:47::  Found all entries for 'Absentee Shawnee Language Class'. Generating XML.
06:50:47::  Starting entry for Aha Punana Leo
...
06:50:47::  Done.


=================================== Stats ===================================
Source: Page Generation Master Directory - Sheet1.csv
        (92.337890625 kB)
Output:
    /Volumes/Home/Documents/Dropbox/ID2050/xmlgen/output/XMLgen-2013-04-30T06-50-47Z-1.xml
        (75.3056640625 kB)
    /Volumes/Home/Documents/Dropbox/ID2050/xmlgen/output/XMLgen-2013-04-30T06-50-47Z-2.xml
        (54.841796875 kB)
    /Volumes/Home/Documents/Dropbox/ID2050/xmlgen/output/XMLgen-2013-04-30T06-50-47Z-3.xml
        (55.126953125 kB)
    /Volumes/Home/Documents/Dropbox/ID2050/xmlgen/output/XMLgen-2013-04-30T06-50-47Z-4.xml
        (48.796875 kB)
    /Volumes/Home/Documents/Dropbox/ID2050/xmlgen/output/XMLgen-2013-04-30T06-50-47Z-5.xml
        (29.6025390625 kB)
    /Volumes/Home/Documents/Dropbox/ID2050/xmlgen/output/XMLgen-2013-04-30T06-50-47Z-6.xml
        (43.333984375 kB)
    /Volumes/Home/Documents/Dropbox/ID2050/xmlgen/output/XMLgen-2013-04-30T06-50-47Z-7.xml
        (17.498046875 kB)
  208 entries in
        |    3 blank entries
```

```
      |  100 mostly complete entries
      |   29 partially complete entries
      |   46 entries missing lots of data
173 content pages out
      |  137 program pages
      |   37 organization pages
      |    0 funding source pages
137 Auxiliary Semantic MediaWiki-related pages:
      |    0 Semantic MediaWiki tag detail pages
      |  105 SMW city detail pages
      |   30 SMW state detail pages
      |    2 SMW country detail pages
      |    0 SMW continent detail pages
 Geocoding was disabled for this run.


   0 SMW tags added to pages


310 Total pages created.
```